DOCUMENT RESUME

ED 107 237 IR 001 977

AUTHOR	Green, Duff, III, Ed.
TITLE	Discovery; New Worlds of Educational Data Systems. AEDS Proceedings, Virginia Beach, Virginia, April 29 through May 2, 1975.
INSTITUTION	Association for Educational Data Systems, Washington, D.C.
PUB DATE	75
NOTE	155p.
AVAILABLE FROM	Association for Educational Data Systems, 1201 16th Street, N.W., Washington, D.C. 20036 ($9.00)
EDRS PRICE	MF-$0.76 HC-$8.24 PLUS POSTAGE
DESCRIPTORS	Business Education; Computer Assisted Instruction; Computer Graphics; *Computer Oriented Programs; Computers; *Conference Reports; Confidentiality; Data Bases; Data Collection; Educational Planning; *Educational Research; *Electronic Data Processing; Facility Guidelines; Higher Education; *Information Systems; Institutional Administration; Instructional Materials; Management by Objectives; Management Development; Management Information Systems; Material Development; Public Schools; Simulation; Teacher Education
IDENTIFIERS	Computer Managed Instruction; DEIC; *Educational Data Systems; OTIS; Planit; TICCIT

ABSTRACT

Proceedings of the Association of Educational Data Systems convention are reported and summarized. Papers are arranged by topic under 21 groupings. Among topics covered are the computer industry's responsibility to education; management by objectives in educational data systems; preparation of instructional material; educational data systems in state education agencies, public-school management, and higher education; teacher training and tools; designing instructional facilities; security in educational data systems; modeling and simulation in instruction, and computers in business education. An author-contributor index is included. (SK)

ED107237

THE ASSOCIATION FOR EDUCATIONAL DATA SYSTEMS

AEDS
Proceedings

DISCOVERY

NEW WORLDS OF
EDUCATIONAL
DATA SYSTEMS

April 29 - May 2 1975
Virginia Beach, Virginia

IR 001 977

U S DEPARTMENT OF HEALTH.
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

Additional copies available for
$9.00, prepaid, from:

Association for Educational Data Systems
1201 Sixteenth Street, N.W.
Washington, D. C. 20036

Library of Congress Catalog Card Number 75-7739

These Proceedings have been made possible by a grant from the IBM Corporation as a service to the educational community.

Printed by Better Impressions, Inc., Vienna, Virginia.

All rights reserved. These Proceedings or any part thereof may not be reproduced in any form without the written permission of AEDS.

Copyright © 1975, the Association for Educational Data Systems

"PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY
Duff Green, III
AEDS
TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER"

3/4

PRESIDENT'S PREFACE

In keeping with the 1975 AEDS convention theme, "DISCOVERY: New Worlds of Educational Data Systems," I hope that the 13th Annual AEDS Convention, especially these convention proceedings, will afford you the opportunity to make many discoveries of your own.

As you know, AEDS was founded to provide a forum for the exchange of ideas and information about the relationship of modern technology to modern education. Our annual conventions, consisting of informational and discussion sessions and vendor exhibits, bring together key people from education and computer technology. AEDS is proud to serve the educational and technical communities in this capacity and will continue its efforts to increase the knowledge and expertise of its members and friends.

This publication is the result of many hours of hard work and should be an excellent supplement to the information received from attendance at the respective sessions. The papers contained within the 1975 Proceedings have been compiled from many that were submitted to our 1975 Program Committee. Each has been reviewed and refereed by a panel of persons knowledgeable in the field of educational data systems, and was deemed acceptable for presentation at this year's convention. AEDS is deeply grateful to each contributor for his or her participation on the 1975 program. It is the Association's desire that this collection of original papers will serve to promote and advance the understanding of educational data processing within the educational community and that this publication will be of assistance to you as a valuable reference guide for years to come.

JAMES AUGUSTINE, JR.
President

EDITOR'S PREFACE

This is the first AEDS convention for which we have had a truly extensive effort in soliciting papers from a broad spectrum of the educational data systems community. You have in these proceedings the selected, excellent responses to our solicitation. We all owe many thanks to the authors who prepared these papers for consideration. On AEDS' behalf I should like to extend particular thanks to the reviewers/referees of these papers: G. Ernest Anderson, University of Massachusetts; Paul A. Bartolomeo, Rhode Island Junior College; Richard A. Bassler, American University; Jacqueline M. Cline, Georgia State University; George W. Gorsline, Virginia Polytechnic Institute and State University; Donald D. Horner, Fairfax County (Virginia) Schools; Barbara L. Kurshan, Salem (Virginia) Community School, James Mitchell, Iowa Department of Public Education; John E. Odom, Mankato State College; Charles K. Rice, Value Engineering; Simeon P. Taylor, NEA; and Carlton Hayes Willis, North Carolina Board of Education.

AEDS is also greatly indebted to Steven M. Raucher, through whom we received the IBM grant which made these proceedings possible, and to Michael J. Healy, who provided invaluable technical advice for the actual printing and publication of the proceedings. I should like to extend my personal appreciation to Gail J. Connelly and Mary A. Horton for their secretarial assistance and especially to Norinne H. Fitzgibbon, AEDS Publication Chairperson, and Charles D.Miller, 1975 Convention Coordinator, for their invaluable support.

DUFF GREEN, III
Proceedings Editor and
Convention Program Chairperson

CONTENTS

I. CAI/CMI UPDATE

Computer Applications in Secondary Education
Arthur J. Korotkin and William J. BuKoski 1

Equalizing Instructional Computing on a Statewide Basis
Kenneth E. Brumbaugh 3

II. COMPUTER INDUSTRY'S RESPONSIBILITY TO EDUCATION

The Computer Industry's Social Responsibility to Education
Richard A. Bassler 7

III. PLANIT USERS SYMPOSIUM

Planit: A Comprehensive Transportable CAI Language
David P. Yens, Charles H. Fry, Cecil Johnson, and Franz Frederick . . . 11

IV. CMI COLLOQUIUM

Individualizing Education Through Computer Managed Instruction (Abstract)
Dennis W. Spuck, Eugene A. Collins, Dean C. Larson, Norman E. Thompson, Donald C. Holznagel, James McNamara, and Ralph Van Dusseldorp 16

V. MANAGEMENT BY OBJECTIVES IN EDUCATIONAL DATA SYSTEMS

Management By Objectives in Developing Educational Data Systems
Edwin E. Blanks 17

Criterion-Referenced Curricular
Decision-Making: Critical Computer Applications
Roger B. Worner 21

Decision Tables in Education
Herbert C. Mayer, Jr. 26

VI. PREPARATION OF INSTRUCTIONAL MATERIALS

Electronic Publishing In Instruction: New Challenge for the Educational Computer Center Manager
Charles A. Morrissey 30

Computer Readings
Edward O. Joslin 37

VII. CAI ROUNDTABLE

Use of Computers for Instruction in the Philadelphia Schools (Abstract)
Sylvia Charp, Henry Altschuler, and Richard D'Orazio 40

VIII. EDUCATIONAL DATA SYSTEMS IN STATE EDUCATION AGENCIES

OTIS: Information System for Oregon Educators;
Instructional Computing for Oregon Students
Jill A. Van't Roer 41

DEIC: New Jersey's System for Bringing Data Elements and Items Under Control
R. Kay Maloney 45

A State Level Education Manpower Information System
John J. Stiglmeier and Michael E. Mirsky 48

IX. EDUCATIONAL DATA SYSTEMS IN PUBLIC SCHOOL MANAGEMENT

Inverted File Structure for Payroll Processing
Thomas C. Richards 53
The New World of Interactive Business Systems
Edward J. Lias 56
An Information System for Enrollment Projection
Robert D. Nelson 64
WICHE/NCHEMS: Implications for Public School Management
(paper not included)
G. Ernest Anderson . . . . —

X. TEACHER TRAINING AND TOOLS

Computer-Augmented Teacher Training: An Undergraduate
Program in Mathematics Teacher Training
Portia C. Elliott and Howard A. Peelle 69
Interactive Repeatable Testing: A Many Faceted Tool
David P. Yens and Victor R. Martuza 74
Student Computer Terminal Supervisors (paper not included)
Gary J. Anderson . . . . —

XI. EDUCATIONAL DATA SYSTEMS IN HIGHER EDUCATION

Applications of Data Processing in the Admissions and Registrars Office
Jack D. Thornton and J. N. Williams. 78
CAA: Computer Assisted Athletics
John H. Hall 83
The Pennsylvania Approach: A Multi-Use Concept for Serving
K-12 and Higher Education (paper not included)
John Nold . . . . —

XII. DESIGNING INSTRUCTIONAL FACILITIES

Improvement of School Design Through the Simulation of
Educational Activities
William S. Bregar. 89

XIII. COMPUTER GRAPHICS IN THE CLASSROOM

Exploring New Worlds
Richard L. Ballard 95
The Use of Computer Modulated Drawing in the Teaching of Art
Charles M. Williams 99
XIV. TICCIT USERS SYMPOSIUM (panel abstract)
Dunstan Hayden, Fred Morrison, and Wanda Rappaport 106

XV. SECURITY IN EDUCATIONAL DATA SYSTEMS

Privacy and the Prudent Computer-Man (abstract)
David Allen Weinberg 107

XVI. MODELING AND SIMULATION IN INSTRUCTION

Model-Building and Computer Simulation for Non-Programming Users
William L. Bewley 108

Use of Computer Simulation in Logistics Management Education
David A. Ameen 115

Teaching Ecology with a Programming Language (APL)
(paper not included)
Anne Pasquino and Howard A. Peelle . . . . —

Throwing Dice by Computer
Alan L. Breitler 118

XVII. ADMINISTRATIVE AND INSTRUCTIONAL COMPUTING IN MINORITY INSTITUTIONS

John W. Hamblen, Gerald L. Engel, Vincent H. Swoyer,
and Lawrence Oliver 121

XVIII. AEDS ROLE IN THE INFORMATION PROCESSING FIELD
(no papers included)

Robert Rector, John K. Swearingen, and Thomas J. McConnell, Jr. . . . —

XIX. HANDBOOK X: EDUCATIONAL TECHNOLOGY (panel abstract)

Ivan N. Seibert, Susan Hawkins, A. Lee Hall, and
James Mitchell 132

XX. COMPUTERS IN BUSINESS EDUCATION

On the Problems of Teacher Preparation in Information Systems
T. C. Ting 133

A New Approach to Teaching COBOL
Joyce M. Forrest 138

Training Programs for Systems Analysts
Harrison S. Carter 143

XXI. EPIC USERS PANEL (panel abstract)

S. M. Raucher, J. E. Ellis, D. L. McReynolds and A. Wood 146

AUTHOR/CONTRIBUTOR INDEX 147

COMPUTER APPLICATIONS IN SECONDARY EDUCATION

Arthur L. Korotkin, Ph.D.
Director, Communications Research

William J. Bukoski, Ph.D.
Research Scientist
Project Director

American Institutes for Research
3301 New Mexico Avenue, N.W.
Washington, D.C., U.S.A. 20016

1. INTRODUCTION

In 1970 the American Institutes for Research, under the sponsorship of the National Science Foundation, completed the first comprehensive survey of computer use in the public secondary schools of the United States (Darby, 1970). Until the publication of this study in 1970 only sketchy data were available regarding the extent and type of use of computers in secondary schools (Goodlad, 1966 and Bangs, 1970). The data from the American Institutes for Research study was published in report and book form.

Since the publication of these earlier data, a virtual void has again developed with regard to current information concerning computer application in secondary schools. Responding to this need once again, the National Science Foundation has awarded a grant to the American Institutes for Research to conduct a follow-up study of computer activities in secondary schools to be conducted during 1974-1975 with a June 1975 completion date. The major thrust will once again be to:

(1) Assess the extent and nature of current computer use in secondary schools with particular emphasis on instructional applications.

(2) Examine and report on selected computer applications which appear to be particularly noteworthy as current examples of the state-of-the-art.

In addition to replicating and updating the previous information, the current study will also undertake to:

(1) Establish some basis for assessing recent trends in computer use (comparing current data with the previous study).

(2) Derive some projections regarding future growth on the basis of the two studies.

(3) Perform some indepth follow-up in those schools where interviews were conducted in the previous study in order to obtain some longitudinal case studies regarding the implementation and institutionalization of computer applications. Such data will be particularly useful in identifying what problems might be associated with implementing computer applications in secondary schools.

(4) Assess the current status and future of computer-based education in the secondary schools as viewed by domestic computer (main frame) manufacturers.

2. PROCEDURE

The data will be collected in a five-step process involving primarily mail surveys and telephone interviews.

(1) A questionnaire was mailed to a 25% random sample of the secondary schools in the United States. A secondary school was defined as any school having any of the grades 9 through 12. Questionnaires were sent to approximately 6,000 schools.

(2) A questionnaire was mailed to every American computer manufacturer. A total of approximately 80 surveys were distributed.

(3) Telephone surveys will be conducted with 25 schools to amplify data collected in the mail questionnaire and to obtain detailed information from schools having exemplary programs of particular interest or impact.

(4) Telephone interviews will be conducted with 10 selected schools visited in the 1969-1970 survey. Data will be obtained of a longitudinal nature to assess changes which occurred to that program since the 1969-1970 survey.

(5) Non-respondent interviews will be conducted by telephone with 100 schools which did not respond to the school survey. The interviews will be conducted to determine if the non-respondent sample was in fact similar to the respondent sample or if some bias may have been operating.

3. RESULTS

Though data collection from all sources--school questionnaire, computer manufacturer's survey, the amplification study, and the longitudinal study--is currently underway, only preliminary findings concerning the school survey can be reported at this time. With an approximate return of 45 percent of the school questionnaires, 52.7 percent of the schools responding were using a computer to assist either in the administration of the school or in direct support of the instructional program, while 47.3 percent of the schools responding were non-users. Administrative uses of the computer included resource management, planning, payroll, personnel records, student accounting (attendance, report cards, schedules), and research/evaluation of teaching methods and curriculum. Instructional applications of the computer included computer assisted instruction, using the computer as a computational (Math, Science), teaching computer science or computer literacy, gaming and stimulation, computer managed instruction, and using the computer in academic and vocational counseling.

In comparing the preliminary data from the current study with the results of the 1970 school survey (Figure 1) we find that the use of computers at the secondary school level has increased substantially. In 1970, 34.4 percent of the schools surveyed were using the computer for administrative and/or instructional uses while in 1974, 52.7 percent of the secondary schools participating in the project have some computer application. Of interest is that the growth of computer use at the secondary school level has resulted because of an 8 percent increase in schools using the computer only for administrative purposes, and a 9.7 percent increase in the number of schools using the computer for both administrative and instructional uses. Schools using the computer strictly for instructional purposes increased by 0.6 percent. Since school computer use is primarily locally funded, it appears that the growth of computers in secondary education is highly related to the demonstrated benefits that the investment will bring to the school or school system.

When compiled the study will provide data which will:

(1) Describe the extent, nature, and purpose of computers in secondary schools; as well as the sources of support, extent of previous use, and plans for future use of computers.

(2) Compare the 1970 and 1974 surveys regarding extent and type of use.

(3) Describe problems encountered in the implementation and institutionalization of educational computer innovations.

(4) Project estimates of future computer use and trends as reflected in the data to date.

Figure 1:
Secondary School Survey:
Preliminary Findings and Comparisons

	1970		1975	
Secondary Schools Using Computers	34.4%		52.7%	
Administrative Only		21.5%		29.5%
Instructional Only		3.9%		4.5%
Both Administrative and Instructional		9.0%		18.7%
Non Users	65.6%		47.3%	
	100.0%		100.0%	

4. REFERENCES

[1] Bangs, F. K. and Hillestad, M. C. (1970). "Automated Data Processing for Education-Curriculur Implications." Detroit: Management Information Services.

[2] Darby, C. A., Korotkin, A. L., and Romashko, T. (1970). "Survey of Computing Activities in Secondary Schools." Washington, D.C. American Institues for Research. R70-13.

[3] Goodlad, J. I., O'Toole, J. F., Jr., and Tyler, L. L. (1966). "Computers and Information Systems in Education." New York: Harcourt, Brace & World, Inc.

EQUALIZING INSTRUCTIONAL COMPUTING ON A STATEWIDE BASIS

Dr. Kenneth E. Brumbaugh
State Instructional Coordinator
Minnesota Educational Computing Consortium

Many challenging, innovative, and worthwhile ways of using the computer as an instructional tool have been developed in recent years. However, only those who are fortunate enough to live in the right location or who are associated with the correct institution have the opportunity to access computer systems for instructional purposes. Local and regional computer networks are being created in many states and have begun to serve users from broader geographical locations. The State of Minnesota has established an organization, the Minnesota Educational Computing Consortium (MECC), whose function includes equalizing instructional computing capabilities for all levels of education in the State. An overview of what MECC is and what MECC has been doing will be presented in this paper.

Minnesota, the twelfth largest state in geographical size, is an educational leader in many areas including educational usage of computers. Such leadership can be documented by the fact that Minnesotans spend $804/pupil unit/year on schools that have an average pupil/faculty ratio of 17.8, and that on any given day have 95 percent of all students present.[1] Support for an instructional computing network comes from the entire educational spectrum, from classroom teachers to the State Commissioner of Education, Mr. Howard Casmey. Mr. Casmey was one of the keynote speakers at the 1974 NAUCAL convention where he presented the Minnesota plan as a model for the delivery of instructional computing services. The history and development of MECC will not be detailed here, except to note that MECC is an organization created by five State agencies according to the rules of the Minnesota Joint Powers Act M.S. 471.59. The five Minnesota agencies are: the Department of Administration, the Department of Education, the University of Minnesota, the State Colleges, and the Community College System.

Principles of Organization and Operation

There are a number of basic principles of organization and operation which, in essence, represent a "constitutional basis" for the joint powers agreement. Among the principles are the following:

1) The funding of services provided by MECC shall be through the member systems which use the services.

2) The funding of communications facilities shall be through a direct state appropriation of funds for that purpose.

3) The policy direction and governance of the consortium will be under the control of the users.

4) The needs for services will be defined by the users through the member systems.

5) Accountability to state government and to the public will be maintained through biennial reports to the Government and Legislature through the HECC, the State Board of Education, and State Information Systems Advisory Committee and by means of the three directors appointed by the Government and Legislature.

6) The consortium must be organized and function so as to maintain and increase the overall computing competence of the educational institutions and systems.

7) The specific instructional, research, and administrative needs will be defined by the systems and must be accommodated by the consortium.

8) The administrative overhead associated with the consortium should be as limited as possible, consistent with needs for effective planning and careful management of limited resources.

9) In general, the service responsibilities to be exercised by MECC will be limited to general-purpose computing.

10) No educational user of computer services and/or facilities shall receive, as a result of joining the proposed consortium, less service or less adequate service than needed and previously available through institutional and system resources.

Responsibilities Delegated to MECC

The member systems of the joint powers agreement collectively have delegated certain powers and responsibilities to MECC which enable MECC to better respond to Minnesota computing needs as defined by the individual educational systems. A partial list of responsibilities is given:

1) Review, coordinate and consolidate proposed biennial computing budgets of the individual member systems.

2) Provide computing services related to the development of educational computing for the member systems including: the review and approval of the acquisition of general purpose computing equipment and systems; establishment of technical standards, including communications; engineering and maintenance of computer facilities; training and consultation services; software development and maintenance.

3) Provide fiscal management of computers on behalf of individual systems of education.

4) Own, lease and operate computer systems. In general, the responsibility by MECC to operate computer systems will apply to all new systems; that is, to those which are acquired after MECC is established. For existing computer systems, including the ownership or leaseing of the equipment thereof; the member could also continue to operate the computer system, under a contract with MECC.

5) Develop long-range plans for computer system hardware, software and services, and for the continued monitoring, modification, and updating of those plans in accordance with changes in technology and in the needs of member systems.

MECC ORGANIZATIONAL STRUCTURE

MECC has been organized into a structure that has staffed three major divisions, they are: Instructional Services, Management Information Services, and Special Projects. The Instructional Services Division has the responsibility of providing and coordinating instructional computing capability within the member agencies of MECC. The Management Information Services and Special Projects activities will not be presented in this paper, however, additional information can be obtained by contacting the MECC office, 1925 Sather Street, St. Paul, Minnesota 55113.

Instructional Timesharing in Minnesota

The MECC instructional timesharing systems serve school districts representing 84 percent of the total school enrollment in the State during 1974-75. Enrollment in districts located outside of the Minneapolis - St. Paul metropolitan area made up only 14 percent of all students having access to instructional timesharing in 1973-74 (prior to the creation of MECC), compared to 46 percent in 1974-75 (the first year of MECC). As shown in Table 1., the instructional use of computers in Minnesota has switched very dramatically from a predominately metropolitan area base to an evenly distributed statewide base. Table 1. exhibits the numbers of students enrolled in school districts having access to instructional timesharing systems in the 1974-75 school year, as contrasted to 1973-74.

Table 1.

ACCESS TO

INSTRUCTIONAL COMPUTER USAGE IN MINNESOTA

ELEMENTARY, SECONDARY, AND VOCATIONAL SCHOOLS*

MECC Region	1974-75 Public School Enrollment Totals	1974-75 Public School Enrollment Totals**	1973-74 Public School Enrollment Totals**
I	81,620	62,377	8,496
II	78,035	67,192	8,070
III	98,54[illegible]	69,968	3.082
IV	70,490	41,499	6.019
V	132,634	85,563	38,160
Metro	399,755	395,471	386,310
ALL	861,[illegible]33	722,070	450,137

* All enrollment figures are taken from the Minnesota Educational Directory, 1973-74.[2]

** In school districts having access to instructional timesharing facilities.

Timesharing Services Provided by MECC

The instructional timesharing services offered by MECC can be grouped into the following four areas: 1) people; 2) hardware; 3) software supports; 4) communications.

People: The most important service that MECC can offer to timesharing users throughout the State is personal response to problems generated by users. The people component of MECC's Instructional Services Division can best be considered as: 1) regional coordinators, 2) support services personnel, 3) operations staff.

In order to facilitate the implementation of computer technology in the classroom, MECC has assigned seven regional coordinators to field offices located throughout the State. The seven MECC regions are illustrated on Figure 1.

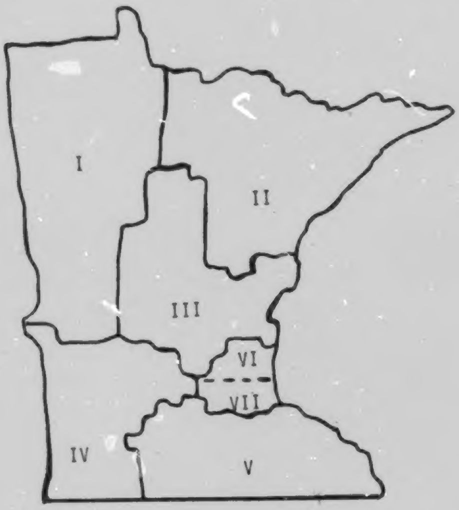


Figure I. MECC REGIONS

The responsibilities of the regional coordinators vary daily from assisting new users signing on the system for the first time to conducting extensive workshops in the development of curriculum materials that use the computer as an integral part of the instructional approach. The second major category for which users can obtain service from MECC is in the area of software and educational support materials. Responsibilities for the personnel working in Support Services include management of system libraries and the writing and distribution of newsletters and instructional materials designed to support instructional computer usage. During 1974-75 MECC is using various computer systems to form the delivery system for timesharing service and is currently purchasing a Univac 1110 system for use in 1975-76. Thus, the operation staff this year is comprised of personnel from the various computer centers, but in early 1975, MECC will be employing its own operations staff.

Hardware: During the 1974-75 school year MECC member school districts are accessing one or more of the following computer systems for instructional timesharing: a Control Data Corporation 6400, Hewlett-Packard 2000 Series computers, a Digital Equipment Corporation PDP-8, or a UNIVAC 1106. Over 800 terminals are located throughout the State and access one of 460 available ports on the computer systems. Table 2 lists the number of ports that are available on the MECC supported systems and where these computers are housed.

In addition to computer hardware, schools need terminal hardware. MECC, using a statewide purchasing agreement, is assisting schools with the purchase and maintenance of this equipment. Over two hundred teletypes have been purchased through MECC by educational institutions in the State this year. Approximately $200 per teletype can be saved by Minnesota educational institutions if they purchase their equipment through the Statewide contract established by MECC.

Table 2.

1974-75 MECC Supported Computer Systems

OPERATING INSTITUTION	TYPE OF EQUIPMENT	NUMBER OF PORTS
Bemidji State College	PDP-8	6
Mankato State College*	UNIVAC-1106	21
Minneapolis Public Schools	H-P-2000C'	32
University of Minnesota**	CDC-6400	256
St. John's College	H-P-2000F	17
TIES***	H-P-2000C'	96
TIES	H-P-2000F	32

* Operated by Mankato State for the Southern Minnesota School Computer Project schools

** Formerly called MERITSS

*** Operated by TIES for TIES member schools only

Software: It is a goal of the Instructional Services Division of MECC to support users by making available to them an extensive library of materials. These materials, most of which are commercially prepared, are available either on loan or at cost. The type of materials kept in stock ranges from programming manuals to instructional packets that assist teachers in using specific computer programs common to the MECC supported computer systems in the State of Minnesota.

MECC personnel are constantly updating the support software by ordering new materials requested by users, and by seeking and converting programs used by other instructional computer systems throughout the nation. MECC users are kept in touch with all the developments of MECC via monthly newsletters distributed throughout the State.

Communications: Establishing a cost effective communications network to the various computer systems is an extremely important factor in making the MECC statewide computer network function, particularly for outstate users. Forty-five participating school districts are within the free-dialing area of the computer system they access. This leaves 198 school districts for which MECC had to establish communication capability. This was accomplished by one of three methods: the installation of foreign exchange telephone lines, the purchase of In-WATS telephone numbers, or by the purchasing of Multiplexors. Forty additional school districts are within the free-calling area of the thirteen multiplexors placed throughout the State. Foreign exchange telephone lines were installed between many communities and this enabled another 114 school districts to make free telephone calls to the multiplexors. Finally, forty-four school districts were given telephone In-WATS capability and this brings the number of total school districts being served by MECC in 1974-75 to 243.

Cost of Instructional Timesharing

All Minnesota school districts desiring access to a MECC instructional timesharing computer during the 1974-75 school year were accommodated. The cost for this service depends mainly on two factors: the total enrollment in the school district and the number of connect hours to the computer. The following six steps outline how the cost to the particular district is calculated:

1) Participating schools have been given a $.60 per student credit to be used for instructional computing time.

2) Computer costs are figured to be approximately $2.00 per connect hour. The subsidy in No. 1 can be used to pay 3/4 ($1.50) of the cost. The district must pay for the remaining $.50. So, the total subsidy divided by $1.50 yields the number of subsidized connect hours for the school districts.

3) The number of HOURS OF SIBSIDY multiplied by $.50 is the school district's minimum cost for computer time.

4) As per the requirement of the legislation that established MECC, a 10 percent surcharge is added to No. 3 above to give the TOTAL MINIMUM CHARGE.

5) For each hour over the SUBSIDIZED HOURS the cost to districts is $.75 plus the 10% surcharge.

6) After a district uses and pays $.75/hr. for a number of hours equal to the number of subsidized hours (see No. 2 above) no additional charge is made and unlimited use is available.

Summary

MECC has been created to serve the general needs for instructional timesharing and administrative service in addition to statewide needs for: 1) computing power, 2) prevention of computer proliferation, 3) cooperation between various levels of education, and 4) equalization of opportunity. An attempt to meet these needs has been undertaken by MECC and an innovative, exciting, and active project is underway in Minnesota.

References

[1] Education Update, July 1974, Vol. 8, No. 6, Minnesota State Department of Education.

[2] Minnesota Educational Directory, 1973-74, State of Minnesota, Department of Education.

THE COMPUTER INDUSTRY'S
SOCIAL RESPONSIBILITY TO EDUCATION

Richard A. Bassler, PhD, CDP, CDE
THE AMERICAN UNIVERSITY
Washington, D.C.

In the quarter century since the advent of the commercially practical computer, the whirlwind pace of technological development and expansion has been increasing. Except for IBM, most companies involved in manufacturing and marketing computers and related products and services have been concentrating on survival. In fact, several years ago, RCA and GE, two giants in the industrial world, found the going too rough in the computer field and abandoned their position in general purpose computer production. Many of the others, especially the smaller companies without strong financial positions, have been concerned with simply staying in business and very few of them even thought about social responsibility.

In the winter of 1973-1974, the author surveyed a number of top executives in the computer field, and asked searching questions. Of 104 companies queried, 54 responded. Based on the annual sales of the respondents and those of the total industry, the response represented 92.4 percent of the industry, with only 7.6 percent not responding. This paper concerns itself with the responses to one of these questions. It hopefully illuminates some of the little known areas with regard to the emerging computer industry's concerns. It examines and evaluates the top executives' thinking and actions about their social responsibilities, especially towards education. The question upon which this paper is based is:

> Do you think that corporations within the United States computer industry, as significant users of the products, the graduates, of our universities and colleges, should have a social responsibility in absorbing some part of the cost of producing this output? If so, why? If not, why not?

This question brought a good response. A few of the respondents avoided the question for various reasons, but the majority demonstrated an admirable approach to the subject.

Proponents of Direct Contributions

Some of the affirmative responses are partially quoted below. A smaller company had this to say:

> . . . American business does have some amount of social responsibility, and this responsibility increases exponentially with size of revenue and profit. The larger a company becomes, the more responsibilty it has to the community and general economy. Basic business decisions by large corporations cannot be made without a least *considering** the overall public good although not to the extent that it might endanger sound business principles and profitability.

In an internal corporate publication, one of the larger companies on the fringes of the computer industry describes its rationale on the company's educational aid program. The document, in part, says:

> More than ever before, the nation's public and private colleges and universities must look to private sources for a substantial portion of the funds they need to maintain instructional programs and their physical plants.
>
> In that light . . . . its educational aid program . . . in the mid-1970's, exceeded $3 million a year and benefitted hundreds of institutions of higher education.
>
> It is a story of a company that *recognizes its need and that of society for the talents and skills of the graduates of these institutions . . . .*

Another company, which does not operate exclusively within the bounds of the computer industry, has a strong feeling about obligations to higher education.

> Although your interest is specifically within the computer industry and we operate in several other industries, I feel that it is important to state that *all of industry has a responsibility to support the cost of higher education. Business and higher education are interdependent. Business is an important source of support for our universities and, conversely, business relies on the universities for trained leadership and growth of knowledge on which technology and innovation depend.*

---

* The italicizing of quoted material from survey responses was added by the author.

Another company, operating almost exclusively within the computer industry, feels that its support should not be restricted to the computer industry. This company's president also feels that corporations *are* supporting higher education through the taxes they pay to the federal and local governments.

A number of larger companies have been giving to education for many years. Some have established patterns of giving, as well as vehicles, for the distribution, thus:

> . . . it has a social responsibility to support higher education. In 1964 the . . . Foundation was chartered to focus such support on four-year colleges and universities, with emphasis on engineering and related scientific education.

Support exclusively to the computer element of education is not envisioned by one corporate president who wrote, "Yes. *This is a normal corporate responsibility in all areas, not just computers.*" Another corporate officer felt that the support of higher education was part of the community responsibility and should apply to all academic areas and not be limited to computer-oriented disciplines.

This broader based support concept is further supported by a vice president of one of the larger companies:

> All major U.S. corporations, regardless of whether they are part of the computer information systems industry, provide financial support to higher education, in the form of contributions to operating budgets, building programs, and scholarships and fellowship programs. *These contributions reflect the firm conviction of corporations that they have a responsibility to support higher education because of its vital importance to the well-being of this country and not simply because colleges and universities are a source of personnel for such companies.* It goes without saying that . . . Corporation is fully committed to this philosophy.

One response in particular recognizes the significant user element as a reason for helping to pay for the end product. This manager of educational relations said:

> Yes, the company does perceive a responsibility (beyond that portion of its taxes which go to public institutions in the states in which we operate). We recognize the cost pressures on higher education, and as a significant "user" of their product expect to help alleviate a small portion of those pressures.

A much smaller company, less than a decade old, with annual revenues less than $20 million, has some strong regional loyalties as well as recognition of only a small sector of the educational community. The president of the company said:

> Yes, but here in . . . we definitely *restrict* our corporate thinking and action to the *privately supported universities, and essentially only at the graduate level.* Why? Because the private institution can generally cover about half its cost through tuition and fees; the other half must come from outside sources, particularly those sources which use the "product" of that institution.

As for assuming responsibility for identifying with the computer segment of the educational process, one respondent took this view:

> We think corporations, including our own, have a social responsibility for support to education *but not necessarily on a quid pro quo basis.*

Related to the *quid pro quo* position taken above, one director of civic affairs had this to say:

> We initiate training programs for specific skills *in which we expect to employ the people involved.* The question of our assuming this same relationship where we have no direct input into the initiation of the courses involved is another matter. It is not customery to assume responsibilities in matters over which you have no control or basic involvement. We are, of course, a major taxpayer, currently assisting in the support of state institutions.

Most of the answers in the responses above were from proponents of direct contributions and were from companies which were established in other fields before the advent of the computer, or which derive a significant portion of their profitability from non-computer-related fields.

Preclusions to Direct Contributions

As might have been expected, a number of responses indicated that the companies had not been able to consider the questions of social responsibility because their attention is focused on the problems of financial survival. The newness of the computer industry and its volatile and dynamic marketplace, coupled with the minimal financing of many of the newcomers to the industry, have created the financial problems.

Comments reflecting the relationships of profitability to philanthropic ability are included here. One company, with an eight-figure loss accumulated in recent years, explained its position:

> Actually, . . . has never been in a profit position where it could consider more than nominal support of educational — or for that matter any discretionary — causes. The company had amassed a retained deficit of . . . at the conclusion of fiscal 1972, and thus *our primary concern has been mere survival.*

Fiscal responsibility to creditors, to say nothing about shareholders, is the theme of the next observation by a vice president of another company with a past record of financial difficulties.

> Although *many of us personally feel that*

> *such support would be beneficial*, . . . is in no financial position to ever consider the subject. Nothing would please our management and Board of Directors more than to have the *luxury of distributing our excess profits to worthy endeavors*. However, we have just rebounded from two very bad years of operations, and *our creditors may not consider such donations to be good business judgment*.

Another new company, but without the serious financial problems of the two just mentioned, is concerned with the problems of keeping the company going on an orderly basis. This company's president observed that: "Considering the fact that we are a new company, it would not be feasibile that we donate to any educational institution monies that we need and presently need in our cash flow."

There is a desire to support higher education, but the means to do so are not presently available, according to two other responses. In one of the letters, the corporate personnel manager says:

> To date . . . growth had required a 'hand to mouth' cash flow. Therefore, since our inception approximately five years ago, we have found ourselves unable to utilize funds for philanthropic purposes of any sort. As we come into our own, I fully expect to establish a pattern of donations to such causes and institutions as consistent with our management's philosophy. The development of this plan has not yet begun.

The other letter states:

> We do not believe that we at . . . have a social responsibility in absorbing the cost of training computer professionals *at this time*. I say this because at this time we do not have the financial means to support such programs. I believe this condition is true of most all the firms presently in the computer industry other than the larger companies (IBM, Honeywell, Control Data, Univac, Digital Equipment) in the industry. . . . smaller companies, such as . . ., are always in need of financing and, thus, not able to support worthwhile individuals in the social area.

The transfer of responsibility to the larger and presumably more profitable companies is the tone of another letter. The vice president does not deny the desirability of involvement, but simply suggests that philanthropy be embraced by the larger companies only. He says that his company probably should be involved but, as a matter of fact, is not.

Opponents to Direct Contributions

A relatively small number of the respondents took the position that their companies and the other companies in the computer industry should not assume any responsibility for education. Their views were expressed unequivocally in concise terms. One response is especially emphatic:

> No! [The] Company should pay market value for work done. Graduates should pay for their 'value enhancement' out of, say, 10% of their wages until cost of education (plus interest) is recovered. . . . My own experience proves it. There are many duties of the corporation as a good citizen. But you must guard against corporate "social responsibilities' corrupting the institution in such a way as to weaken it in performing it's [*sic*] far more noble and beautiful roles in creating customers, jobs, and profits.

A similar view was taken in another response in which the company president said, "I pay as part of the salary I pay for college graduates." He also commented that "Contributions do not add to profit and therefore work against this plan." "No, . . . thinks students and states should. Corporations are already overburdened," said another respondent, a vice president of a company claimed by him to be too small to be considered in this survey.

Two other responses have generally the same tone. One says, "No! The responsibility and mission of corporations is to produce goods and services! Not Graduates."

The second says, "No! As taxpayers we all (corporations and individuals) support education. I don't think universities should train 'products' for industry and industry shouldn't 'order' people produced.

Not favoring direct support by the members of the industry, the next two respondents suggest support from government and from the entire United States population, respectively. Their comments are partially quoted as follows:

> . . . I do feel that higher education is best served by direct government support or self-funding. Corporations should fund higher education insofar as their best interests are affected.

* * *

> I think that it is the responsibility of the entire United States population to support our colleges and universities and, indeed, all of the educational facilities within the country. I do not think that the computer industry should be singled out as a special (albeit significant) user of this resource. It is the entire country that benefits from education and the entire country should support it.

Arguments that the stockholders would oppose an expenditure for higher education were limited. One letter states, "No, the stockholders would never agree that this is the proper conduit for getting appropriate monies from one place to another." Another viewpoint reveals the attitudes and interests of the stockholders:

> Until two years ago, our company did not have a formal budget plan concerning corporate gifts. It was Mr. . . .'s philosophy at that time that public corporations should not be in the position of giving away what properly belonged to the shareholders. I

> would also say that this philosophy was molded by the fact that he *personally* gave, and continues to, a great deal of his resources through his own philanthropy . . . . a formal program has now been established with the primary giving going to organizations in support of the greater . . . area.

Summary and Analysis of Responses

Over half of the respondents are exercising their social responsibility as contributors to higher education. While denying social responsibility, another respondent gave this opinion:

> I don't think that the corporations in this case have a social responsibility, but I do believe that *they have a very valid quid pro quo interest in maintaining a strong academic discipline in computer sciences.* This would lead me to comment that for *those corporations in the computer business, support of this academic area should have a high priority on the list of academic disciplines to be supported* through the corporate contributions effort.

Only 16.7% of the respondents took the negative position. Their reactions were stated emphatically. Some responses were so couched in doubletalk that it was exceedingly difficult to place them in a suitable category. The relationships of the responses are shown in Table 1.

TABLE 1

ANALYSIS OF RESPONSES
TO SURVEY QUESTION ONE

	Number	Percent
Yes, do have responsibility and do provide support	19	35.2
Yes, have some responsibility . . .	9	16.7
Have not thought about it because of poor financial condition	6	11.1
No, very positively	7	12.9
No, but have an interest in further consideration	2	3.7
Too small to be concerned about the problem	2	3.7
Not answered in tabulatable form . .	9	16.7
	54	100.0

Conclusions

The computer-related industry is not very different from the rest of American industry and business with respect to philanthropy. Having long-established patterns of giving, the major companies generally provide assistance to education in the same proportion each year. In some cases, the pattern of giving is modified to conform with changing demands. Three of the largest organizations are often responsive to federal spending and are more likely than not to change contributions to higher education as required. The consensus seems to be that the total amount to be donated would remain essentially the same. Only the distribution of funds would reflect a modified pattern of giving.

The overall picture of social responsibility to higher education within the computer industry is affected by companies operating on a marginal financial basis. Formany of them, the impact of federal aid to education and the rate of philanthropy is at the bottom of any list of priorities. For others, there is a lack of time and personnel for researching the desirability and practicality of possible philanthropic programs to aid higher education.

Regardless of what the concept is called, the idea of providing financial assistance to institutions of higher education makes good sense. Henry Ford, quoted in a *Fortune* (May 1973) article makes a distinction between social responsiveness and social responsibility. "I think that they are quite different," he explains, "We need to be socially responsive without overdoing the social responsibility — in the sense that we've got to do something directly about the deficit in the school system, for example."

-o-

PLANIT

A COMPREHENSIVE,TRANSPORTABLE CAI LANGUAGE

A Symposium sponsored by

The PLANIT User's Group

Chairman: David P. Yens, University of Delaware

Participants: Charles H. Frye
Northwest Regional Educational
Laboratory

Cecil Johnson
Chief, Systems Integration and
Command Control, U.S. Army
Research Institute for the
Behavioral and Social Sciences

Franz Frederick
Associate Professor
Department of Education
Purdue University

David P. Yens
Educational Services Consultant
and Assistant Professor
College of Education
University of Delaware

INTRODUCTION

PLANIT is a sophisticated, comprehensive, and transportable computer assisted instruction (CAI) author language that has been under development for over nine years. PLANIT has been installed at over 30 university, military, and industrial locations. The purpose of the symposium is to provide information and discussion concerning the features, implementation, and uses of this powerful language that has been described as one of the three seminal CAI systems in existence today. Yet, the most exciting feature of PLANIT is that it can probably be installed on _your_ computer with relative ease.

The first presentation, by Charles Frye, describes the capabilities of PLANIT and its past, present, and future. Cecil Johnson will discuss several implementations of PLANIT on military computers. Current uses of PLANIT and several examples of available courseware will be reported by Franz Frederick. David Yens will be concerned with research uses of PLANIT and the means of evaluating courseware. Summaries of the presentations are given below.

A REPORT ON PLANIT, VERSION 3*

Charles H. Frye

The development of the machine transportable version of PLANIT (Programming Language for Interactive Teaching) was begun in 1968 under contract to the National Science Foundation. PLANIT is an instructional system consisting of an author language and supporting computer programs for preparing, editing and presenting any subject matter suitable for individualized presentation to students within the constraints of the communication equipment that is currently available. It provides:

1. A wide range of authoring conveniences for fast and efficient preparation of lesson material
2. A comprehensive calculation utility allowing both the author and student to use a natural language to access the computer
3. Necessary support features for time shared presentation of the lessons to students
4. Automatic collection of data and maintenance of progress records from session to session.

Maximum portability has been a prime objective to permit installation on a variety of different computers with relative ease.

Work on PLANIT has proceeded at the Northwest Regional Education Laboratory since December, 1972, to develop a version of PLANIT which could be regarded as a "production" version in that it is free from known errors and includes all necessary support features for day-to-day computer center operation. The current version 3 of PLANIT has now been delivered which seems to meet these requirements.

As an operational system, PLANIT contains far more than just a facility for authoring and dispensing CAI lesson scenarios. In fact, it is a complete time sharing system specifically designed for this particular application. It has run on

---

* The PLANIT work is supported by NSF grant number EPP73-07319 A04.

computers which support no other time sharing work. It has run along side of another time sharing system in the same computer. And it has run as a subordinate program to a time sharing system as well. The procedures used for installing PLANIT permit a high degree of flexibility. In addition to the expected conveniences for preparing instructional materials, PLANIT offers many other needed and/or desirable services, such as:

- File management and several levels of file protection
- Several options for enhancing continuity from session to session
- Complete backup facility using magnetic tape
- Sophisticated error detection and automatic recovery with minimum loss of users' work
- Prompting aid always available at the terminal at any reasonable requested level of detail
- Automatic file handling on both disk and tape for inexperienced users
- User accounting and billing
- Operator console interface allowing complete control of the system with minimal burden to the operator
- Completely flexible overlay configurations featuring "look-ahead," making the system adaptable to available core space while maintaining the fastest possible response time.

The report examines the significance of three major aspects of the PLANIT project: to the user, to computing centers, and to programming technology. The first, the significance to the user, is due to at least four facts:

1. The PLANIT language is very easy to learn initially, requiring less than an hour's orientation, but incorporates a sufficiently large repertoire to allow the user to increase his sophistication gradually to very powerful capabilities.
2. The expense associated with installing the system is low enough that it usually does not require a major commitment on the part of the computer center and can often be implemented at the request of a relatively small group of the faculty.
3. A Users Group exists for PLANIT which sponsors a quarterly newsletter to more than 50 paid subscribers (at a subscription rate of only $6.00 per year), aside from many more on their mailing list.
4. PLANIT is operating or being installed at more than 20 universities, several computer corporations, and several military installations, with the latter proposing to expand to several hundred systems. PLANIT lessons are completely compatible among installations.

Secondly, the significance of the project to computing centers is due mainly to:

1. The availability of time sharing instructional software capable of sophisticated CAI applications at a trivial initial cost which can be installed by personnel who are normally available within the center.
2. The adaptability of the system that will accomodate to any reasonable operating environment.
3. A system which is designed to add only the minimum necessary burden to the workload of the computer operator, mainly the mounting of tapes when off line backups are desired.
4. Complete automation of system features, where possible, of such things as user protection, error recovery, user accounting and billing, etc., but also allowing the operator to override these features manually when necessary.

In the case of programming technology, the PLANIT project is significant in that it provides a model of the first fully-transportable time shared operating system. It is one answer to "everybody talking about transferability but nobody doing much about it." The PLANIT systems for all of the various installations (and most major venders' products are involved) are generated from the same physical set of source code statements, virtually guaranteeing full compatibility. This truly permits an "unbundling" of software costs from the expensive hardware.

Finally, the report will describe the expected direction of future PLANIT efforts, with the intent to expand the system to accomodate multi-terminal communication for sophisticated simulations and game playing involving many players. These kinds of CAI applications are still in their infancy because of the scarcity of systems on which to experiment. Military projects have taken the lead in these areas with their man/machine command and control systems but it is expected that the technique can have a much broader application.

Implementation of PLANIT at the U. S. Army Research Institute for the Behavioral Sciences*

Cecil D. Johnson

This paper will describe the experience of the Army Research Institute for the Behavioral and Social Sciences (ARI) in installing, using, and evaluating PLANIT as a CAI language in an Army setting. Plans and aspirations ARI has for utilizing PLANIT for both operational and research applications will be provided to explain why ARI has wanted to implement PLANIT. ARI interest in PLANIT has been threefold: (1) as a laboratory with a research mission in computer assisted instruction (CAI) with a goal of evaluating PLANIT itself; (2) to use PLANIT as a research tool across the full range of the mission of ARI in the behavioral sciences; and (3) to aid interested members of the Army training establishment in developing experience and competence in CAI.

---

* The views expressed in this abstract are those of the author and do not necessarily reflect the views of the U. S. Army or of the Department of Defense.

It is hoped that the recounting of the problems ARI had in achieving its pioneerihg implementations will not deter anyone with a need for PLANIT from attempting its installation. In my opinion, a man knowledgeable of the system software of any computer system with a FORTRAN computer and a word length of at least 24 bits should be able to install PLANIT in a week--if he has the assistance of a programmer who has installed PLANIT on any computer, same or different. Without such assistance the documentation for installation is sufficiently scanty as to require a longer effort.

As the Army developing agency for training and educational technology research, ARI has participated in the development and/or evaluation of several computer-based instructional systems. Thus, ARI can view PLANIT from the perspective that comes from having considered a variety of solutions to many diverse training problems.

ARI interest in PLANIT largely stems from: (1) the relative ease of installation on computers of widely different architecture, (i.e., its transportability advantage); (2) its easily learned but powerful author language; (3) the economy with which courseware can be produced; and (4) the ease with which courseware that has been produced (i.e., authored) on one kind of computer can be executed by students or trainees (or trained staff members with respect to proficiency maintenance) on a quite different kind of computer system. The clear cut advantage of PLANIT over all other CAI author languages on these four dimensions made the possible superiority of other languages with respect to the cost of a student hour of instruction, special terminal features, and/or graphics seem relatively unimportant for many Army training applications.

Initially, the Michigan State University version of PLANIT (a version developed from the same starting point as was the Purdue/NSF PLANIT version) was installed on the ARI CDC 3300 computer. Many modifications were made, some because the ARI computer, unlike the MSU CDC 6500 system, had a very small core memory (i.e., 32K 24 bit words), some because of the limited interactive features inherent in the executive system in use, and many in an attempt to make PLANIT reflect the original documentation without the constraints that the MSU time sharing system has imposed on some of them.

The second installation was on a tactical field data system. This installation involved the coordination of courseware to be developed by the Systems Development Corporation with software installations by ARI personnel. Several difficulties were encountered that required field modifications, but PLANIT was installed and running by the target date. Because of some differences from the NSF/Purdue version of PLANIT, this version was designated Version 1.1.

ARI has research reports describing the results of the initial experiment designed to see if effective CAI courseware could be installed on a field data system. It is sufficient for the purpose of this paper to say that the CAI trained groups were superior when the subject matter was tactics or weapons but results were ambiguous for the groups trained in high school level mathematics. Overall, it could be safely said that an effective CAI system had been installed in a short time frame on a tactical data system. The objective of the ARI project had been achieved with positive results.

ARI also has a terminal equipped with a card reader and printer and a number of CRT and teletypewriter terminals connected by dedicated telephone lines to a Univac 1108 computer located at Edgewood, Maryland. The computer aided (i.e., interactive dialogue supported by a data base) career counseling project is the primary ARI application utilizing the 1108 computer during the early fall of 1973. PLANIT has been demonstrated at Fort Sill, Oklahoma; Fort Leavenworth, Kansas; Fort Monroe, Virginia; and from the West Coast using the Edgewood 1108 system.

ARI has conducted studies on the ease with which PLANIT can be learned. CAI lessons on PLANIT were produced in the PLANIT 1.1 language for use on a PLANIT controlled system. This product was tested with research psychologists and military personnel. It appears that research psychologists can become productive, but limited authors, in about 10 hours of on-line instruction and can complete the additional courseware covering the more sophisticated, but less essential, courseware in another 15 hours.

The Army has a family of field data systems that have the same central processor, the Litton 3050 (or ANGYK-12). The compilers in general differ for the different systems even though they use the same central processor with differing amounts of core memory, types of secondary memory and configurations of peripherals.

The development of embedded training packages (ETP's) for systems containing the L3050 would be greatly facilitated if PLANIT could be installed on each tactical data system. ARI has a project underway to install a full power PLANIT on one of these systems (i.e., TACFIRE) using an approach which will permit the same executive system and associated compiler to be used across all tactical field data systems containing the L3050 computer.

In November, 1974 ARI assembled a team to install (in three days) PLANIT 2.5 on the CDC 6500 at Fort Leavenworth. One of the team had previously visited McMasters University to assist (for less than a week) in the installation of PLANIT 2.5 on the CDC 6400 at that location--as preparation for the installation of PLANIT at Fort Leavenworth. It is anticipated that the ARI field unit at Fort Leavenworth will make considerable research use of PLANIT and that both the Command and Staff College and the Combat Arms Concept Development Activity at Fort Leavenworth will use PLANIT respectively for teaching and for the examination of interactive information systems requirements.

The ARI experience with implementation of PLANIT has convinced many of us in ARI that PLANIT is a highly superior, readily learned and economically used, authoring language, is highly transportable, and is a very practical CAI language for use in accomplishing many Army applications.

CURRENT USES OF PLANIT, COURSEWARE AND COURSEWARE CONVERSIONS

by Franz J. Frederick

Because of a current lack of detailed information concerning instructional uses of PLANIT and PLANIT installations, the national PLANIT USER'S GROUP is preparing survey instruments to collect these data.

The data to be collected for installations deals with mainframe descriptions, numbers of terminals in use, scheduler control, PLANIT version in use, I/O rates, types of terminals, parameterization values, and special modifications in use or anticipated. The data to be collected for courseware includes content descriptors, student level, objectives, prerequisite knowledge, adjunct materials, lesson type categorization, lesson description, use of specialized language features such as WAIT, TIME, C:, computed GO TO, RECYCLE, TEXT, etc., the number of characters per line and lines per display (screen) allowed, and any restrictions on the availability of the lesson materials.

The lesson type categories currently being considered are: (1) question/answer series with branching, (2) graphic or tabular constructions, (3) learning games, (4) simulation, (5) automated testing, (6) programs which learn (AI), (7) student controlled sequences, (8) author training materials, (9) conversational simulations and (10) a catch-all category called "others".

In spite of detailed information it is known that PLANIT has indeed been used successfully for production CAI both here and abroad with a broad range of subject matter. It is reasonable to say that the primary use of PLANIT at many installations is that of experimentation, lesson development, and/or controlled research in learning.

Because of the richness of the PLANIT author language and the PLANIT CALC language, some of the current uses of PLANIT include: (1) conventional CAI lesson presentation, (both linear and branching), (2) generative CAI lessons, (3) experimentation with CMI algorithms, (4) automated testing, (5) simulations, (6) learning games, (7) conversational simulations, (8) demonstrations of basic examples of artificial intelligence, (9) tabular graphics, (10) student-controlled sequencing.

Because of the general transportability of PLANIT, considerable interest has been generated in using PLANIT at sites where other CAI languages have previously been used. This was the case at Purdue and in fact materials had been designed or acquired for five different languages. These were Coursewriter I, Coursewriter III, PICLS, FOCAL, and BASIC. An anlysis of these languages and PLANIT revealed eleven levels of language conversion problems. These eleven levels reflected difficulties of implementation ranging from the trivial to not possible through author code to systems level conversion effort. The eleven levels ranged from direct subsitution of one operator for another to directly accessing host computer system functions.

Overall estimations of time to convert a line of author code were developed for several levels of difficulty of conversion.

The various lessons converted ranged from 30 lines to about 1100 lines of author code. The resulting conversions required PLANIT lessons ranging from 5 to 35 frames in length.

Estimations of proportions of level of conversion difficulty for a lesson by languages were developed. Several features which could not be converted were identified.

In the case of PICLS lesson conversions, a large number of lessons existed and an effort was made to design a SNOBOL translator for PICLS to PLANIT conversion. Generally the PICLS lessons posed conversion problems which were about 60 percent trivial and 40 percent non-trivial The translator conversions accomplished nearly 70 percent of the necessary code conversion leaving only about 30 percent to be converted by hand.

If numbers of lessons in the same language need to be converted the design and use of a SNOBOL translator can be a very useful conversion technique. The next several years should see many lessons converted to PLANIT thus promoting a broad scale sharing of usable CAI materials.

PLANIT: RESEARCH USES AND COURSEWARE EVALUATION

David P. Yens

PLANIT is a computer assisted instruction (CAI) system that brings powerful and exciting opportunities to both the curriculum developer and the researcher who do not have access to special computer systems. For the instructional developer, the capabilities of PLANIT provide powerful tools for the preparation, evaluation, and modification of the instructional content of a program.

The method of curriculum development that is recommended by many authorities for optimizing learning is the systems approach which involves the following steps:

1) Analysis of the learning task
2) Preparation of behavioral objectives
3) Determination of content sequencing
4) Construction of the program
5) Testing of the program
6) Analysis of the results
7) Revisions if necessary; go back to step 4
8) Implementation of the program

Although PLANIT will not help with steps 1 and 2, the author assistance features and extensive student records system will greatly facilitate the remaining steps. The program content can be changed in response to observed student performance characteristics, changes in the curriculum field, or a desire to modify the course. Detailed evaluations of student performance at any point in the course are encouraged by the records-keeping and summarization capabilities of PLANIT. This same process permits authors to evaluate postcourse results in terms of specific performances by students on relevant sections of the computer-based material.

For the researcher, PLANIT provides precise stimulus control and manipulation, complete response recording, timing of responses, and individualized modifications of stimulus presentation sequences based upon individual responses. It also permits the random assignment of students to different experimental conditions that are administered via PLANIT.

The paper will describe several opportunities, advantages, and examples of using PLANIT for research and for the evaluation of instructional developments. Below is a brief summary of several areas of concern.

PLANIT is designed to optimize computer-human interactions. This would encompass such areas as CAI, completion of employment questionnaires, counselling interviews, vocational exploration and testing, and on-line training. For all these applications, it is important that the student or user interpret the author's communications in some desired way. For instructional use, the author must insure that the student learns the desired information provided in each section of the program. Formative evaluation procedures involve the use of evaluation data to modify, revise, and improve an instructional program during development. With PLANIT, the author should have students or users try out each section of the program as it is developed. Student records data for each student and summary data on all students for each frame and concept can be analyzed to determine the way users responded to each frame and/or associated sets of frames. On-the-spot modifications can be made to the program based on these data. This evaluation-revision loop may be repeated as many times as necessary to produce the desired learning.

Once the course or program is complete, summative evaluation procedures are used to determine the overall effectiveness. Both cognitive and affective domains may be of interest. In fact, several CAI programs administer a final examination and an attitude questionnaire at the end of a program so that all pertinent information will be in the records file for each student. Again, student records data can be used to determine students' paths through the program to assess their performance on each frame or segment. The results of the final evaluation can be related to specific intra-program performance characteristics and modifications may be made again if required.

Several fundamental questions must be asked before any courseware development takes place using PLANIT. For example:

1) Is the use of the computer for the specific application a valid one? Does it provide instruction or perform some other function that cannot be duplicated by other methods with equal effectiveness at lower cost?
2) Is the value or effectiveness of the program worth the developmental and operational costs?
3) Is the response time of the computer terminal adequate to keep students interested? What is the desirable response time for the content to be administered?

Several aspects of these questions and their implications for the selection and use of PLANIT will be discussed.

The research applications of PLANIT are almost limitless. Different instructional strategies can be incorporated into different versions of a program and their relative effectiveness compared. Many types of verbal learning studies, from paired-associate learning to complex sentence or paragraph learning, can be done with PLANIT. Special types of studies in which changes in stimulus materials are contingent upon student responses are particularly well suited to this medium. The measurement and control of response times by PLANIT makes possible many types of research concerned with response latency; however, this kind of research requires a computer which can accurately measure terminal response times.

Research in which students are to be assigned to different groups based upon some measurable characteristic can be automated with PLANIT. The prgram could administer the measurement instrument, assign the student to the appropriate group, and present the appropriate treatment.

PLANIT has a powerful calculation facility, making research on mathematical variables attractive. For example, the difficulty of a mathematics task could be adjusted to maintain the same apparent difficulty for each student without regard to mathematics ability or knowledge. Of course, an infinite number of different types of arithmetic problems could be randomly generated by PLANIT for both research and instruction purposes. PLANIT can solve the problems and evaluate the students' answers.

Brief examples of these ways in which the computer has been used for research will be reviewed in the paper and the applicability of PLANIT for future research will be discussed.

In summary, PLANIT has great potential for instruction, research, and other uses requiring interactive communication with people. Pretesting of materials and their easy modification based upon student records analysis by appropriate evaluation methods is one of the major advantages of PLANIT. If properly and conscientiously performed, the evaluation techniques will greatly assist in the development of quality programs, and it will be on the basis of quality programs and products that CAI and PLANIT will continue to grow as a respected method of instruction and medium for research.

INDIVIDUALIZING EDUCATION THROUGH COMPUTER-MANAGED INSTRUCTION

Chairperson: Dennis W. Spuck
University of Wisconsin - Madison

Presenters: Eugene A. Collins and Dean C. Larsen
Jefferson County Public Schools,
Colorado

Norman F. Thompson and Donald C. Holznagel
Minnesota School Districts Data Processing
Joint Board

James McNamara
Wisconsin Research and Development Center

Discussant: Ralph Van Dusseldorp
University of Iowa

Programs of individualized education have been implemented in a large number of the nation's schools. Such programs focus on the individual needs of students and allow students to progress at their own rate in accordance with their needs and abilities. Increasingly, too, students are able to make their own decisions concerning objectives and subject areas to be studied. It is evident, however, that programs aimed at individualizing education require pupil accounting and decision-making patterns which demand the use of computers in instructional management.

Each of the participants in this symposium is currently involved in the development of an interactive CMI system, designed to assist in student record keeping and to provide teachers with information to be used in prescribing or guiding the instructional process in individualized programs. Eugene Collins and Dean Larsen will describe their experiences in building a computer model of an existing instructional system from within a school environment and will identify the advantage of an on-line approach in reducing interface problems which can exist between technical and user systems. Donald Holznagel and Norman Thompson of TIES will provide information concerning their work on the Individualized Learning Activities (ILA) system. Development of the Wisconsin System for Instructional Management (WIS-SIM) will be described by James McNamara. In addition to a general overview of each of these three systems, the presenters will focus a portion of their discussion on problems and concerns related to prescriptive and grouping functions in CMI systems. Such functions supply information to teachers to assist them in selecting appropriate educational experiences and settings for students enrolled in individualized instructional programs.

The session will begin with an introduction to Computer-Managed Instruction, presented by the symposium chairperson. Following the presentations, Ralph Van Dusseldorp will comment on the papers, and his comments will be followed by open discussion.

MANAGEMENT BY OBJECTIVES IN DEVELOPING EDUCATIONAL DATA SYSTEMS

Edwin E. Blanks
Virginia Commonwealth University

The management of an urban university is faced with many of the challenges facing sister institutions of higher education. Located in the heart of a metropolitan area with commuting students, commuting faculty, scarcity of real estate, a concrete campus, parking headaches, and so on, we find administrators constantly assessing viable alternatives to insure organizational continuity and direction. Performance planning along with the ubiquitous communication channels are two important factors affecting the interaction of the individual with the organizational structure and conversely the organizational structure with the individual. Management by Objectives is not the panacea for all university wide administrative functions. With proper direction, adequate feedback mechanisms, and executive support the MBO Program simply will not be just another way to control institutional activities. It will become the formidable method for the overall accomplishment of messhing specialized support functions into an influencial, functional unit.

Why is management by objectives popular? Are analogous management techniques or methods of management replaced? What is the prime mission of the organization? These are simple questions on the surface. However, many administrators left alone to determine guidelines, plans, and propitious use of existing resources find it a difficult task to develop workable answers to the aforementioned questions. Hopefully this paper will develop some understanding of the MBO Program in developing Educational Data Systems, highlight several barriers encountered in implementing successful programs, and attempt to answer the previously mentioned questions.

There are many overwhelming reasons why a new management program is unrealistic at any level in the organizational hierarchy. The numerous daily pressures, the coordination of component departments, and deviations from the norm make it difficult to practice the management functions of planning. organizing, directing, and coordinating. Also many managers possess the inclination that MBO is only designed for the 'top' management of an organization. Comparison of several companies with successful MBO programs has depicted the most effective application of this approach is located in the ranks of middle management. The secret of their success hinged on the participation by both managers and subordinate personnel in developing the program.

The term management by objectives is often applied to just the organizations operating in a dynamic environment where timely information handling is responsive to both internal and external needs. Also the term MBO may present a semantic problem for some organizations and individuals. Objectives, goals, plans, targets, results, performance appraisals are just several expressions used interchangeably in the MBO context. Individuals stressing absolute preciseness in establishing and defining objectives detect a fundamental weakness in attempting to provide a specific MBO statement. As we all suspect, definitions are strickly personal. As utilized in developing educational data systems, my definition of the term management by objectives is defined as: "A management technique stressing the results to be achieved while allowing involvement of the individual to establish specific criteria by which he will be expected to perform." Once again it is important to emphasize the personal nature of definitions.

There are several keys to the success of an MBO program. Most analysts and programmers are more responsive to adhering to their own personal objectives than they are to objectives imposed on them by management. Large developmental staffs as well as smaller ones can attain job enrichment and increase productivity by personal involvement in the MBO program.

Unquestionably, the term MBO could be construed to replace the traditional merit review. This misconception can be also applied to the use of objectives in preparing organizational budget programs. Institutional planning of one, three, and five year programs is commonly referred to as the MBO programs. Departmental responsibilities utilizing greater self determination and encouraging positive attitudes among participants will align with the MBO program at the detail level. Simply stated, whatever the program and its use, and if the program has not been mandated from the executive level, objectives stressing results are here to stay.

PROJECT ORIENTATION

Close supervision of professional analysts and programmers is no longer tolerated at any level in the life cycle of systems development and implementation. New employees entering the field are better educated, in some cases highly motivated, and possess technical expertise some managers have never been provided. The manager is still responsible and accountable but finds it more difficult to apply the management questions of what, why, when, and how. Assuming present trends, the size and complexity of the developmental effort is constantly increasing. The myriad time consuming activities associated with developing an educational data system places the manager in the difficult position of not being thoroughly aware of all of the detailed diversified activities of his staff. Compiling modules, catalogued procs., copy library specifications, program library maintenance, invoking JCL, data base concepts, TSO, and a host of other specialities formulate the manager's awareness of the fact that his accomplishments are really the individual results of many specialized support functions working as a unit. The manager is applying the management functions of planning and organizing these specific technical areas to attain the desired results defined in the MBO program.

The university organizational structure consists of two divisions, academic and medical with a provost on each campus. Functional vice presidential areas, namely Finance, Student Affairs, and Planning and Operations transcend both divisions. The Director of Computing Activities reports directly to the President. There are four departments included in the realm of our university computing environment and the systems and programming effort is one of three sections in the Administrative Services Department.

The systems development effort is organized along the lines of project teams. The three teams are each headed by a systems development supervisor (project leader) and have systems analysts and programmers assigned specifically to the team. The academic team handles admissions, registrar, administrative school functions, alumni, and continuing education applications. The financial team accomplishments deal with payrolls, cost accounting, budgetary system, student accounts receivable, and other supplemental financial sub-systems. The third and last major group concentrates on inventory applications, training, documentation, standards, structured walk through procedures, and controls. The existing structure which handles both new development and maintenance activities fits into a results oriented framework. Some of the individuals remain task oriented but for the past 18 months most of the analysts and programmers have adapted the self development approach in our MBO program.

MANAGEMENT BY OBJECTIVES SYSTEM

Generally, implementation of the MBO program requires an understanding of the management functions and clearly stated objectives. Making objectives clear is not always an easy task. Objectives are only clear when they are well stated. The criteria for making objectives clear are:

1. Individual - Directly traceable to a person doing his job
2. Measurable - The ability to quantify identified indicators of progress
3. Specific - Delineates the particular area of performance targeted for improved results
4. Realistically-Set - Based on sound analysis not wishes

Objectives are clear only when the above criteria exists and when they are understood and mutually agreed upon. Objectives may be established in any one of the three following ways:

1. Imposed from the top
2. Jointly evolved
3. Initiated by the subordinate

The jointly evolved method normally has the best chance for success. Here the manager and the analyst or the analyst and programmer sit down and jointly agree on the objectives and the results to be attained. Regardless of approach, answers to several questions must be developed. What resource support does he need to meet the objectives? Will these objectives affect other employees or other stated objectives?

The acceptance and implementation of a successful MBO system is contingent on indicating the areas targeted for improvement before the fact. This establishes a firm line of communication, a mutual understanding of what is expected, and a results orientation necessary to remove the personality contest from the performance review cycle. How is this accomplished?

We have found the use of an application item Check List and Plan List to be the accepted vehicle. The Planning List requires only a yes or no indication next to each application item required for successful development of any assigned project. The application items have associated with them individual standards. These standards are succinct statements located in the standards manual and used as a guide for the analyst and programmer in completing his work. The list of application items consists of many traditional systems components. Such things as flowchart, data element dictionary, program narrative, program listing, procedure manual, file specifications, sort/utility statements, and catalogued procedures. This list is not all inclusive and items can be added or deleted to suit your organization.

There is also a technical support section to be completed prior to beginning work on the project. This section provides for control in maintaining the validity of the test and source modules. The checking of both areas is normally handled during the structured walk through. The use of this valuable tool, the plan and check list document, allows the MBO system to function in the chronological sequence of determining what must be done at project initiation. This provides both management and the analyst/programmer the "before the fact", mutually agreed on, objectives necessary to complete the assigned work.

PLAN LIST Required Item Write Yes or No	APPLICATION ITEM (STANDARD NUMBER)	CHECK LIST Approved By Sign & Date
	Feasibility Proposal (308)	
	Documentation Index List (320)	
	Implementation Schedule (316)	
	Application Flowchart (328)	
	File Specification (336)	
	Sort/Utility Statements (360)	
	Program Narrative (408)	
	Program Flowchart (412)	
	Program Listing (416)	
	Catalogued Procedure (384)	
	Job Execution Deck (388)	
	Procedure Manual (380)	
	Data Management Manual (724)	
	Keypunch Binder (728)	
	Test Data (364)	

Sign & Date ______________ Approves the above assignment of required items.

COMPLETED BY TECHNICAL SUPPORT

Approved By
Sign & Date

COPY LIBRARY

______________ • File specifications added - changed - deleted in SYS1.COBLCOPY. Circle if appropriate.

SORT/UTILITY CONTROL STATEMENTS LIBRARY

______________ • Sort/Utility Control statements added - changed - deleted in SYS1.SORTPARM. Circle if appropriate.

PROGRAM LIBRARIES

______________ • Production status module which needs to be modified, has been moved from SYS1.LIBMSTR to LIBRARY.SOURCE.

______________ • Module has been moved from TESTLIB to SYS1.LOADLIB and LIBRARY.SOURCE to SYS1.LIBMSTR.

______________ • Module has been scratched from TESTLIB and LIBRARY.SOURCE.

CATALOGUED PROCEDURE LIBRARY

______________ • Application procedures have been added - changed - deleted in SYS1.PROCLIB. Circle if appropriate.

This technique can be utilized throughout any phase of the empirical analysis in the systems development cycle. Where the developmental aspects may consist of such things as problem definition, preliminary survey, systems planning, conceptual design, detail specifications, and technical requirements, the concept using objectives can be applied. It will also work in the areas of programming, client training, implementation, and ongoing maintenance. The rationale for discussing the many areas in which objectives might apply is to help broaden the parochial view of many task oriented managers, analysts, and programmers

BARRIERS

As with any new approach or new system there is always the possibility of encountering formidable barriers. Certain individuals in the organization will have definition problems. They begin to play the word game. The use and misuse of specific terms while defining these terms creates a situation where the employee states he does not understand the program.

Another barrier is the extremely busy employee. The routine and the important duties of this employee in general is a common occurrence in many organizations. The employee may lack job flexibility or he may represent a one man system, either of which constitutes a barrier in implementing a successful program. He sees this as just more red tape to contend with and not an intrinsic portion of his prodigious duties.

Finally, there are many employees who regard this system as too difficult to understand. This often happens where the MBO program is being mandated from the top by edit. The employee does not recognize this as something he wants to identify with or learn. In some cases this is the old, overworked resistance to change probia. Forcing this change will normally not be successful. Giving the employee the opportunity to chat and ask questions about the program is one way of possibly gaining his confidence. There are many other obstacles in the path of installing and implementing an MBO system. Only several of these barriers have been indicated.

SUMMARY

The concept of Management by Objectives in developing educational data systems is a challenging and exciting one. The material presented in this article is provided to spark your appetite. Let each project area or each department establish their own objectives and then discuss the planning cycle necessary to implement these objectives. The MBO system utilized in the systems and programming developmental effort is a valuable tool and when used properly will increase productivity.

REFERENCES

E.E. Blanks, Application Documentation - A Life Cycle Tool, Paper presented at CAUSE, Washington, D.C., 1974.

Dick H. Brandon, Data Processing Organization and Manpower Planning, Petrocelli Books, 1974.

J. Daniel Couger and Robert W. Knapp, Systems Analysis Techniques, John Wiley and Sons, Inc., 1974.

Gordon B. Davis, Management Information Systems: Conceptual Foundations, Structure, and Development, McGraw-Hill, 1974.

Glen H. Varney, Management by Objectives, Dartnell Corporation, 1971.

Management by Objectives, Association For Systems Management, 1971.

Managing Improvement Through Performance Objectives, Practical Management Associates, 1973.

CRITERION-REFERENCED CURRICULAR DECISION-MAKING: CRITICAL COMPUTER APPLICATIONS

by Dr. Roger B. Worner
Executive Director
Institute for Educational Research
Downers Grove, Illinois

The track record of curriculum personnel in harnessing the speed and versatility of the computer for curriculum decision-making purposes has been most unimpressive. And there should be little wonder about this. In the early years of computer installation in school systems, it was the business-oriented educator who perceived the promise and opportunity that computer technology held for processing data in innumerable versions and combinations to enhance fiscal decision-making and accountability. Curriculum leaders continued to be among the entrenched skeptics who believed that anytning really worthwhile should be humanly processed with a careful measure of reflectiveness and, most undubitably, should not be left to the clumsy machinations of an awkward combination of integrated circuits controlled by an unimaginative computer programmer, who most frequently spoke in a foreign language of bits, bytes, core, disk, mod, and other unintelligible verbiage. So the world of the curriculum educator progressed in the age of computer technology, and even today, most school district instructional leaders having access to the computer degrade its magnificent processing capacities to the level of a class III clerk typist. In this light, there should be little wonder that the priority rating of most instructional applications in all but a few school districts falls only slightly behind the production of the Expenditure Report on Janitorial Supply Utilization.

Fortunately for us curriculum people, a small but increasing number of instruction-minded educators are now envisioning the dynamics of computer technology and are wresting control of a valuable aid - lost by default - that will furnish the key to critical curriculum decision-making in the public schools. Now that the retainer walls have given away, it would appear that an endless flood of "new" curriculum and instruction questions of crucial significance to the operation of school districts are being raised. Among them are "Can the computer be of assistance in sequencing instructional skills and processes taught in the classroom? Can it help in organizing curricular materials and rating them for classroom usage? In what manner can it assist in more accurate student diagnosis, placement, prescription, assessment, supplementation, grouping, and monitoring? Can it help to specify inservice training needs?" The number and quality of the questions are heartening. Curriculum people are arriving, finally. At the same time, however, they are not infrequently finding that the present degree of curricular specification is not sufficient to enable the computer to provide meaningful, usable information that can be applied to sound curricular decision-making.

The first order of business, it would appear, is for curriculum leaders and designers to prepare themselves for the demanding data requirements requisite to complex curricular decision-making. When this is at a sufficiently accomplished state, the computer will provide analytic responses with the sophistication desired to dramatically improve instructional services for students, teachers, and administrative decision-makers.

Preparation for Curricular Decision-Making

Making effective curricular decisions through the application of a computer requires thoughtful planning to identify the priority areas of a school district's informational needs, the scope of those needs, and the practical and potential utility of information that is gathered or generated. A curricular specification effort will surely follow this planning and will be directly dependent upon these factors. It is an axiomatic reminder that the computer can not and will not respond to questions which have not been asked, and its analytic responses will rarely be more sophisticated and have greater utility than that which was originally perceived by the inquirer.

Several of the most critical areas where the computer can be applied to curricular decision-making are (1) test generation, (2) student diagnosis, (3) student placement, (4) student grouping, (5) student prescription, (6) student monitoring, (7) skill and process sequencing, (8) curricular material organization, (9) select curricular material usage, (10) curricular material revision, (11) identification of inservice training needs for staff, (12) achievement gain parameters, (13) cost/effective analysis, and (14) curricular material warehousing and deployment. In most school systems, a few of these functions are being performed. Yet, all of them can be effectively accomplished in school systems with or without the use of a computer, though the rapidity of processing and the potential combination of analyses are unquestionably enhanced by computer availability.

The degree of curricular specification required to enhance decision-making in each of these fourteen areas is not nearly as prodigious and time-consuming as one might initially imagine. At the same time, it is noteworthy that the required specifications are so basic to the operation of curricular and instruction programs that they should have been accomplished in all school districts whether or not there is an interest in addressing any of the suggested strategical decision-making areas. Minimal specification activities to prepare for computer-applied curriculum decision-making are (1) the identification and sequencing of instructional skills and processes, (2) the referencing of identified instructional skills and processes to available curricular materials in the school district, (3) the construction of representative test items for each instructional skill and process, (4) the referencing of developed test items to each instructional skill and process, and (5) the establishment of a series of decision rules that will be applied to data generated by the computer.[1] Unquestionably, there are an excessive number of additional, more detailed curricular specifications that can be undertaken to improve school district curricular content, delivery, and assessment, but it would appear prudent to obtain, initially, the wealth of curricular decision-making information that can be derived with a modest time and cost input. With that successful experience realized, the need for and desirability of forging ahead in more detailed areas of curricular investigation and specification become readily apparent.

Identifying and Sequencing Skills and Processes. Primary to curriculum and instruction decision-making of any type is that school districts clearly specify the detailed foundations upon which their programs are based and the instructional elements that the programs convey to the student. Whether school districts label the instructional foundations for their programs as skills, processes, concepts, elements, principles, or a combination of these terms, it is vital that they be explicitly identified for each program, grade level, course, or subject. Furthermore, the instructional foundations will need to be sequenced in a manner that seemingly best represents a logical pattern for introducing and instructing these elements to students as they progress through a course, grade level, or program.

The process of identifying and sequencing skills and processes is not a particularly awesome task, though it is doubtlessly flawed because school district personnel have rarely attempted it before. At the same time, however, every textbook, workbook, and aid used in the schooling process has a stated sequence, and contrary to what many people will claim, most instructional staff members follow the publisher's sequence quite religiously. Thus, in the absence of any more sophisticated approaches, staff members selected to identify and sequence skills and processes can accomplish the process by replicating publisher skills and sequences or empirically deriving them.

Referencing Skills and Processes to Curricular Materials. A more time-consuming but no more difficult task to prepare for curricular decision-making is the referencing of instructional skills and processes to available textual, workbook, and aid materials frequently used by classroom teachers in conveying skills and processes to students. The process amounts to citing the title of a book (workbook or aid) and the page numbers where a particular instructional skill or process is treated. This task will need to be done for each skill and process in each program, grade level, course, or subject available in the school district or, if a lesser approach is selected, to those programs, grade levels, courses, or subjects on which the school district wishes to gather and process decision-making data.

Obviously, the more extensively that skills and processes are referenced to a broad range of available school district curricular materials the greater will be the decision-making potential to school administrators and classroom teachers.

Constructing Representative Test Items. A task crucial to curricular decision-making is the construction of representative test items for each skill and process identified in each program, grade level, course, or subject. Since the test items will be used innumerable times in collecting deciding information, it would be advisable that test item writers be trained in the intricacies of performing this task. The process is not difficult to learn, and highly successful results can be achieved with a minimal amount of capable guidance.

School districts may wish to consider the development of several test items (perhaps as many as ten) for each skill and process to enhance the capability of generating multiple forms of tests with varied test items.

---

[1] Detailed examination of these and additional specifications for curricular decision-making are found in Roger B. Worner's, Criterion-Referenced Diagnosis, Placement, and Prescription (Nashville, Tennessee: Learn, Inc., 1975).

Referencing Test Items to Skills and Processes. Since test items are developed for explicit skills and processes previously identified, referencing the items is a simple clerical matter. In the process of carrying out the item to skill and process referencing, it is worth noting that because of the previous skill and process referencing to curricular material, the school district will have accomplished a test item to curricular material referencing without the need for actually undertaking that process.

Establishing Decision Rules. At some point in the conceptualization of each of the curricular decision-making areas, school district personnel will need to establish decision rules that govern how the computer should process, act upon, and report data. As an example, if placement tests for reading are generated and administered to first grade children, rules will need to be established governing how the computer should correct the test; indicate skill/process proficiency, deficiency, or questionable mastery; establish a student's placement based on deficiency; document the type and number of potential instructional prescriptions that may be used for beginning or follow-up instruction; and report other information in a form and according to specifications desired by administrators and teachers.

The setting of decision rules is not a difficult process, and the only certain guidelines for setting them are that the user have confidence the rules will not distort the data in a manner which would cause questionable or faulty decision-making to occur. Thus, those who establish decision rules have the responsibility to assess the degree of stringency that will be applied to data. If decision rules are too stringent, they may over-discriminate; if they are too lax, they may under-discriminate. Under any condition, however, decision rules - like skills, processes, sequences, test items, referencing, and the curricula itself - are changeable, and a part of the rationale for undertaking these activities is to learn. As such, the setting of decision rules will need to be done with care, but the expectation that it will be done without error and revision is unrealistic.

Criterion-Referenced Curricular Decision-Making. Having completed the specification activities for curricular decision-making, school districts will have delineated criterion-referenced data, information that is built upon or reflective of actual local programs and procedures. The value of criterion-referenced data specification for decision-making is that it captures the essence of the content, sequence, and measurement criteria of local programs, while non-criterion-referenced data specification, at best, only approximates what "may be" the substance of local programs and, at worst, fails to reflect local programs at all. The obvious advantages of criterion-referenced data specification are that the data acquired have a direct bearing on the content, sequence, procedures, material, and measurement that are being employed in the school district, and the decision-making data collected cannot be dismissed or rationalized as irrelevant. Once that barrier is removed, school administrators and teachers can look objectively and with confidence at gathered data and ferret out curricular areas which would appear to need attention. They can also proceed with a "controlled" adjustment of content, procedures, materials, and measurement variables - from a knowledgeable vantage point - in attempting to effect improved curricular programs.

Critical Computer Applications in the Curricula

The critical computer applications in the curricular area are those which effect the accurate and effective delivery of services to students. These applications revolve around the detection of a student's instructional status (test generation, student diagnosis, student placement, and student grouping); content delivery (student prescription and student monitoring); adjustment of curricular content and delivery (skill and process sequencing, curricular material organization, select curricular material usage, curricular material revision, identification of inservice training needs of staff, and curricular material warehousing and deployment); and broad-based, curricular decision-making (achievement gain parameters and cost/effectiveness analysis).

Test Generation. The prior specification of school district skills and processes and the development of test items referenced to those skills and processes enables curriculum personnel to use the computer in the generation of a multitude of different forms of criterion-referenced tests for diagnostic, placement, and assessment purposes. Depending upon the parameters established by school district personnel, tests can be constructed to diagnose or assess any phase of programming or student learning: multi-grade level, single grade level, multi-discipline, single discipline, skill strand, unit stand, and numerous others.

Student Diagnosis. Using tests generated on the skill and process content of a single or series of grade levels and applying decision rules reflecting the degree of a student's proficiency (e.g. mastery, non-mastery, questionable mastery), curriculum personnel can use computer technology to diagnose the location and degree of student proficiency and deficiency.

Student Placement. Diagnostic information derived from criterion-referenced tests can be used to ascertain a student's instructional placement position in a program, grade level, course, or subject. By establishing and applying decision rules to the frequency, succession, or pattern of skill deficiency identified in criterion-referenced testing, the point at which a student's knowledge of the curricular program breaks down can be established, and the point of beginning instruction can be determined.

Student Grouping. Since school district personnel are able to acquire diagnostic and placement data on individual students using computer applications, they are also able to apply these individual datum to groups of students and determine the frequency, composition, and range of student deficiency for the purpose of initiating instructional groupings. Such a procedure can be applied on a classroom, grade level, program level, or school level basis depending upon the desires of school administrators and teachers.

Student Prescription. The referencing of test items to school district skills and processes, and, in turn, the referencing of the latter elements to curricular materials, permits curriculum and data processing personnel to construct computer applications that will yield information on student prescription. Each skill or process included in a curricular program is represented by one or a number of sets of instructional materials which can be used to convey the meaning and content of the skill or process. Depending upon the decision rules and parameters set up to report prescriptions, the computer can either report all (or some) identified prescriptions for the student's placement skill, the same information for all skills in which the student has exhibited deficiency prior to his placement position, or both types of information.

Student Monitoring. School districts can employ the computer as a periodic or continuous instructional monitoring device through on-line or batch process modes. Monitoring can be used to ascertain student progress, reassess grouping patterns, affirm the presence or absence of proficiency or deficiency, test out the efficacy of prescriptions and original placement, and perform any number of administrative and teaching evaluations of curricular content delivery. It is noteworthy to mention that on-line processing and instantaneous teacher and student feedback, promoted by some as requisite for successful instructional activities, is most likely neither cost effective nor necessary if other curricular activities (e.g. skill and process identification and sequencing, curricular material specification, referencing, and criterion-referenced test construction) are properly and completely carried out.

Skill and Process Sequencing. School district personnel can obtain valuable information from the computer with which judgments about the appropriateness of instructional skill and process content and sequences may be tendered. Among other information yielded by computer-generated criterion-referenced tests are "deficiency counts" on each skill and process conveyed in the instructional program. Deficiency counts specify the incidence with which a given skill is answered incorrectly by students and states when there is no relationship between the incorrect answers and the student's final placement. If a skill's deficiency count is high - on a variety of different test items - it may suggest that the skill is inappropriately placed in the sequence or, it in some instances, is of questionable importance to the curricular program.

Curricular Material Organization. Just as the computer can be used to rearrange and extract skills and process from a curricular program because of their inappropriateness, it may be used in an identical manner to cause the rearrangement or deletion of curricular materials. Since curricular materials are referenced to instructional skills and processes during the curricular specification process, curricular materials may be resequenced by manual or machine reprogramming at the same time as each skill and process is being resequenced.

Select Curricular Material Usage. Perhaps among the more difficult tasks facing classroom teachers is selecting and employing curricular materials with children exhibiting forms of exceptionality (e.g. learning disability, emotional problems, educable mentally handicapped, blind, deaf and hard of hearing, and others). Because instructional centers for exceptional children are frequently scattered throughout a school district and the nature of the teaching task is inherently more difficult, teachers of the exceptional child require a greater degree of feedback on the instructional efficacy of curricular materials used with students having very unique learning problems. Such feedback may be obtained by recording the specific prescriptions employed with the district's exceptional children in mastery of skills or processes. The storage and processing of such data on a district-wide basis can be used to aid in strengthening the school district's curricular holding of select types of materials, preparing prescription profiles for exceptional students, projecting the characteristics of materials which have a high probability of success, deleting questionable materials, and mapping instructional strategies.

Curricular Material Revision. The computer is a valuable tool in assessing the points at which curricular materials are weak and in need of revision. By collecting feedback from teachers on prescriptions used in instructing specific skills and processes and assessing student criterion-referenced test results, evidence can be collected which suggest the need for a careful analysis and possible revision of curricular materials. Particularly is such analysis of value in assessing materials used to convey skills on which students frequently exhibit deficiency and which are likely to affect student placement.

Identification of Inservice Training Needs of Staff. Skills and processes that are frequently found deficient in post-criterion-referenced test assessment (after the student has received instruction) serve as a potential basis for curricular material revision, inservice training of staff, and, more likely, both. Again, the computer can serve as a data retriever and analyst in determining the frequency and identity of vital, yet troublesome, skills and processes with which students exhibit a moderate or high degree of deficiency even after instruction has been carried out. Such information may be used to structure training sessions on methodologies, materials, and techniques for conveying skills and processes that are directly linked to student failure in the school district.

Curricular Material Warehousing and Deployment. Using end of the year diagnostic, placement, and prescriptive information gathered from computer-generated and processed criterion-referenced tests, a warehousing and deployment plan may be established and implemented to specify the exact quantity and type of district curricular materials delivered to each building and classroom in the school district. Such a plan insures that classroom teachers will have material resources readily at hand on the first day of school and that they will be specifically selected on the basis of each individual child's assessed needs. The classroom teacher may then apply diagnostic, placement, grouping, and prescription information on an individual or group basis immediately without experiencing the delays and errors that are characteristic of material deployment at the beginning of a school year.

Achievement Gain Parameters. School districts that have undertaken basic curricular specification will determine that the computer has unique capabilities for gathering and analyzing achievement data. For example, based on pre- and post-test administration of school district criterion-referenced tests, achievement gain data can be collected by school, program, grade level, department, or other organizational unit. Such information can be used to assess the priorities for funding, staffing, material allocation, and resource deployment to schools, programs, departments, and grade levels where unique learning problems appear to be causing underachievement. Flexible achievement gain parameters may be established - based on the past performances of school district students - to assess the present and project the future efficacy of instructional programs in delivering appropriate services to students. Though such parameters will always be limited by the fact that the tested student clientele is forever changing, they have the inherent capability - when combined with other information - of identifying sources of curricular design or programming that are weak and require further study or immediate change.

Cost/Effectiveness Analysis. School districts employing a planning-programming-budgeting system framework for gathering and analyzing cost data are in a unique position to apply those data to achievement information gathered through curricular specification and derive cost/effectiveness comparisons among instructional programs. Using cost figures derived from a program budget and achievement gain figures acquired from the pre- and post-administration of criterion-referenced tests, cost/effectiveness ratios can be documented for similar programs, courses, and subjects. Determinations can then be made on the effect of cost increases, decreases, or stability on the variable effectiveness of diverse instructional practices, procedures, or support systems. School districts will find that cost/effectivness procedures are particularly valuable for assessing the relative merits and practices of alternative programs that are being considered for implementation, especially when cost is a vital consideration. Numerous other valuable insights are provided through the derivation of cost/effectiveness analysis. Among them are critical information for long-range planning, goal setting, priority setting, budget deliberation, innovative program design, and many others.

Conclusion

The practical applications to which the computer can be applied by school district personnel is virtually limitless. It would appear, however, that the most promising and significant utilization of computer technology in school districts may be in the domain of curricular decision-making. With a minimal amount of internal curricular specification - involving the completion of tasks that are so basic to school district operation that they should have been completed decades ago - an entire panorama of improvement activities can be initiated to aid school districts in increasing the quality of curricular services they deliver to students. Perhaps the most significant aspect of computer utilization for the schools' cautious curriculum leaders to recognize is the potential that this tool holds for humanizing instruction for students and refining the content and make-up of curricular offerings. Without imposing any content or procedural limitations on school teachers and administrators, it can aid the schools in viewing the student in his fullest complexity and help to design an endless number of alternatives that can help him to achieve both personal and academic success in the schooling process.

DECISION TABLES IN EDUCATION

Herbert C. Mayer, Jr.

IBM Corporation

One of the frequent problems discussed in user meetings covers the sharing of programs or applications between installations. Due to the different languages that may be used as well as the varying internal structure of the hardware of the computers, transfer of programs has required almost a total reprogramming effort to implement an application at a different installation.

Starkweather (1969) indicated his concern about the transfer of instructional material between computers when he stated:

> The language should be versatile and inclusive so that other languages in which extensive curriculum materials have been developed can be translated into it. The language processor itself should be written in a readable computer programming language to aid its implementation on a variety of computers (p. 272).

Zinn (1969) analyzed many instructional languages and addressed the translation problem by stating:

> New languages and systems will have greater capacity for translation of instruction programs from present programming languages in which they were implemented. Translatability is possible without imposing any restrictions on innovative ideas for language or strategy (p. 33).

Feldhusen (1970) in his review of CAI research and development made this recommendation concerning an instructional language:

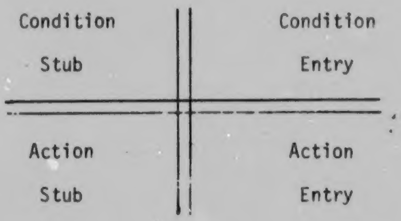
> For the present the diversity of languages and the relative complexity of most languages stands in the way of program development and precludes transfer from system to system (p. 7).

One way to assist with the transfer of information from one system to another is to use decision tables in many of the educational applications of the computer. First, let us discuss the concepts of decision tables and how they are used.

A decision table or logic structure table is a tabular representation of a procedure for expressing the logical relationship between a set of conditions and the actions to be taken when specified combinations of the conditions exist. There are four major parts to a decision table formed by intersecting vertical and horizontal double lines. According to Montalbano (1974), each part has its own specific function and two of these are designed to answer the following questions:

1. What factors have to be considered before a course of action can be determined?
2. What actions are governed by this procedure?

The answer to the first question is found in the upper left corner - the condition stub - as shown in Figure 1. The answer to the second question is represented by the lower left section of the decision table - the action stub. To the right of the double vertical lines is the entry part, which in turn is divided into the condition entry above the double horizontal lines and the action entry that is below the dividing lines. The condition entry contains answers to the questions asked in the condition stub, while the action entry consists of specifications of the actions described in the action stub. The three different types of tables depend on what kind of entries are used in the table. A limited entry table means that the answers in the condition entry and the specifications in the action entry are generally limited to YES or NO. When more general entries, such as words or phrases are used, the table is called, extended entry. Tables which combine these two types, where entries across a row are consistently of one type, are called mixed entry tables. The entry sections of a table tie it together and provide the basic difference between decision tables and other forms of procedure descriptions.



Condition Stub	Condition Entry
Action Stub	Action Entry

Figure 1. Decision Table Arrangement

The difference is the manner in which the branching structure - more commonly called the program logic - of procedure is displayed. Most procedures include conditional actions that are performed only if a set of conditions holds true. Program logic is this combination of inter-relationships between conditions and actions. In narrative descriptions and flow charts, which are the most common methods of describing program logic, the conditions and actions are intermixed. By contrast, in decision tables, they are separated, with all the conditions listed in the condition stub section and all the actions specified in the action stub.

To illustrate the use of a decision table, consider the following situation taken from Article II of the Constitution of the United States:

> No person except a natural born citizen, or a citizen of the United States at the time of the adoption of this Constitution, shall be eligible to the office of President; neither shall any person be eligible to that office who shall not have attained to the age of thirty-five years, and been fourteen years a resident within the United States.

When the information in this narrative form is placed in a decision table, the conditions for eligibility are easier to comprehend. All of the possibilities are clearly delineated in the decision table shown in Figure 2. The Y stands for YES, N stands for NO, and X indicates the action to be taken. In the action entry a blank corresponds to no action for this activity. The blank in the condition entry represents a "don't care", "ignore", or "it doesn't matter" situation regarding a particular condition. Further explanation of the use of the blank in an entry will be shown in the next example.

Presidential Eligibility	1	2	3	4
Age 35 or over	Y	Y	Y	N
Natural born citizen	Y	Y	N	
Resident of U. S. for 14 years	Y	N		
Mark eligible for President	X			
Mark ineligible for President		X	X	X

Figure 2. Presidential Eligibility

The right part, or entry section, of a decision table is divided by vertical lines into columns called rules, with the numbers at the top of the column identifying the rule number. Each rule includes both a condition part and an action portion, thus connecting one to the other. The interpretation of a rule, using Rule 1 in Figure 2 is as follows: If a person is 35 years old or more, and a natural born citizen, and a resident of the United States for at least 14 years, then that individual is eligible for the office of the President of the United States. At any condition where the individual fails to qualify, indicated by N, the result specifies that the person is ineligible for the office. Everything written in the narrative form in the Constitution is neatly laid out in this table, which permits a quick and easy check of an individual's eligibility.

In general, a rule is a conditional directive of the form, "If the checked conditions listed in the condition part hold, then take the actions specified in the action portion". For purposes of computer processing, the information contained in the decision table is usually coded in symbolic terms that are easily handled by the computer. Thus, the Yes or No entries in the decision table can be represented by 1 or 0 respectively, which are symbols that can be processed by the computer easily. The person who prepares the original decision table need not perform this coding, since this activity can be handled quite readily by the computer using a specific program. Condition entries in a limited entry table, thus lend themselves to simple conversion by a computer program. On the other hand, entries in an extended entry table, consisting of words, phrases, or expressions, frequently require the inclusion of a code book or conversion aid, in order to assist the computer in setting up a processable form.

Consider another example used by a college admission officer to determine a student's acceptability for entrance to the college according to the following criteria:

> All students with a grade point average of 3.0 or better will be admitted. Those students with grade point averages between 2.5 and 3.0 and an SAT score of 800 or better will be admitted. A student who is first in his high school graduating class will be admitted.

Arranged in decision table format, the information would appear as shown in Figure 3.

This college admission table is an example of a mixed entry table. The condition entries for the first two conditions use phrases which represent an extended entry type table. The remaining entries, for the third condition and all the action entries, are of the limited entry type. All entries across a row must be of the same type, either limited entry or extended entry. The blanks in some of the condition entries indicate that a particular condition can be ignored, or it does not matter or pertain to the rule. Thus, a "Don't Care" answer, represented by the blank, is equivalent to ignoring a particular condition. Or, stated in different terms, the answer to the condition with the blank response does not affect the rule so that it doesn't matter in the overall consideration. Most decision tables will contain "Don't Care" entries, otherwise the tables would be unmanageable in size. A decision table with eight conditions would require a total of 256 rules to express all possible combinations if there were no blank entries. The ELSE rule illustrated in Figure 3, permits the collection of all the cases that do not meet the conditions specified by Rules 1, 2, or 3. This rule provides a considerable saving in expressing the various combinations permitted in the table.

The general considerations about decision tables and their representation can be summarized as follows:

1. All the questions that should be asked or conditions to be tested are found in the condition stub.
2. All the types of actions to be taken are found in the action stub.
3. The answers to the questions or the conditions to be tested for a given procedure are

College Admission	1	2	3	Else
GPA	3.0 or better	between 2.5 & 3.0		
SAT score		800 or better		
First in H. S. Class			Y	
Admit to college	X	X	X	
Send sorry notice				X

Figure 3. College Admission

found in the condition entry and constitute the condition portion of a rule.

4. All the permissible actions for the procedure are found in the action entry and constitute the action portion of a rule.
5. Rules set apart by vertical lines that define columns are defined by the correspondence between sets of conditions and courses of actions.

An example will be given to illustrate the use of a decision table with specific subject matter. Consider the case of an algebra teacher who is teaching the concept and evaluation of quadratic equations using the quadratic formula. The decision table in Figure 4 indicates the part of the formula that separates the evaluation of the given expression into rational, irrational, or imaginary roots. It is true that each case can be solved by using the quadratic formula, but simplified evaluation can be accomplished by the use of factoring. This breakdown not only uses more than one method, but shows the student where and when to apply different types of solution strategies. The table will now be explained in terms of decision table concepts. The situation should be analyzed in terms of the conditions and actions involved in the activity. When these have been elaborated the relationship between these conditions and actions should be set up in the form of rules.

Quadratic Formula	1	2	3	4
b^2-4ac is	>0	>0	=0	<0
Is b^2-4ac the square of a number?	Y	N		
Evaluate by factoring	X		X	
Evaluate by formula		X		
Evaluate imaginary roots by formula				X

Figure 4. Quadratic Formula

The first condition stated in the table concerns the value of the expression b^2-4ac, the discriminant, which in the quadratic formula is the quantity inside the square root symbol. For positive values it is helpful to check another possibility (condition 2) to see if the number is the exact square of a number. In this case the easiest solution can be obtained by factoring the original equation. If it is not the square of a number, the roots or solutions of the equation will contain radical signs and the exact values of the solution can only be approximated. Rules 1 and 2 cover the possibilities discussed so far. Rule 3 provides a special circumstance for the situation when the value of the discriminant is zero, and this situation yields two equal roots, which can be found easily by the factoring method. When the discriminant is negative, the resulting roots are imaginary and can only be found by using the formula. The first condition listed in the first row across the table is an example of the extended entry form of entries. Any suitable expression or words may be used to specify what is needed. The second condition uses the Yes - No type entries of the limited entry form. These are easily expressed in the computer program as a 1 or 0. Since for Rules 3 and 4 the answer to condition 2 does not affect them, no entry is made in the corresponding columns. The required action for each rule is indicated by the entry X to specify what action should be performed.

As an aid to teachers who would plan to use decision tables in connection with their educational program development, the following steps are listed for decision table preparation.

1. Identify the problem to be studied.
2. Obtain or develop a statement of the process or procedure that describes the technique.
3. Pick out from this statement the conditions that must be met and place them in the condition stub.
4. Pick out from this statement all the actions that must be taken and place them in the action stub.
5. Develop the rules that prescribe the actions required for each possible combination of conditions. This information is specified in the entry portion of the table.
6. Check to determine that the table is complete and the rules are not contradictory.

Several comments are appropriate in connection with this development process. For some applications there may be no written set of procedures that give an indication of what should be done. This deficiency may be corrected by interviewing the people involved in the activity to develop the needed written procedure. Once the statement has been prepared all the conditions and actions that are contained in the procedure can be selected and listed in the stub portion of the decision table. Setting up the relationship of the conditions and actions through the rules will follow the concept of: if this set of conditions holds, then these are the actions to be taken. The final step of checking the accuracy of the table can be handled by some computer programs that will expand the table to its maximum size for a fixed number of rules, then reduce the rules to eliminate all redundancies. In the development of the table the teacher may find that it is getting too large. If this is the case, sections of the process can be placed in separate tables and the tables can be linked through an action statement at the appropriate point. This technique helps to keep the tables at a manageable level.

Computer processors for decision tables have been available for some time. In recent years where there has been any activity at all with decision tables, the emphasis has been to shift the processing activity from the batch-controlled environment to one that can be handled through input at a terminal. The programs that will not only check the accuracy of the table and execute it, will also permit entry of the table information through the terminal and guide the preparation of the table itself. The biggest problem that the computer programmer faces is the integration of decision tables with the procedures that have been coded in a programming language. The language of implementation of the decision table processor may be different from that used for other parts of the computer instructional program. The problem may be corrected by letting the operating system control the change between languages or by writing both parts in the same computer language.

Due to the varied makeup of different computers, a computer program will not run on two computers with different internal characteristics. Even though processors exist for common computer techniques, the translation of material on one computer to a second computer depends either on a translation program to allow the computer to perform the change or on processors on different computers that can handle common input, such as a decision table. The exchange of information between computers cannot be accomplished without the specific computer programs and processors. By using the decision table processors that are currently available on different computers, the process of transferring instructional material or computer data in decision table format from one computer to another can be considerably simplified.

References

1. Feldhusen, J. A position paper on CAI research and development. Palo Alto, California: Stanford University, ERIC Clearinghouse on Educational Media and Technology, 1970.

2. Montalbano, M. Decision tables. Chicago: Science Research Associates, Inc., 1974.

3. Starkweather, J. A. A common language for a variety of conversational programming needs. In R. C. Atkinson and H. A. Wilson (Eds.), Computer-assisted instruction: A book of readings. New York: Academic Press, 1969.

4. Zinn, K. L. Implications of programming languages for mathematics instruction using computers. In R. T. Heimer (Ed.), Computer-assisted instruction and the teaching of mathematics. National Council of Teachers of Mathematics, 1969.

ELECTRONIC PUBLISHING IN INSTRUCTION:
NEW CHALLENGE FOR THE EDUCATIONAL
COMPUTER CENTER MANAGER

Charles A. Morrissey, Executive V.P.
Time Share Corporation

Some of you may recall the experience of the young sociology graduate who traveled to Egypt to study the changing status of the wife in Egyptian family life. One status symbol was particularly significant, that of having the wife walk behind the camel while the husband rode.

The sociologist discussed this trait with many of the men and suggested that the woman's role was more important and should be recognized.

Not long after these discussions she was watching one large group making their way into the desert and noted that all the women were walking in front. Feeling she had stimulated a significant change in the culture of this tribe she asked one leader, "Don't you feel that this recognition of women is deserved?"

"No," he replied.

"Well why are you having the women walk in front?"

"Land mines," he replied.

My recent experience has shown that managers of educational computer centers are facing a similar transition - from a traditional role of supplying generally standard administrative needs to the educational user to a new era - that of managing instructional software and other multi-media. This new area which I, and many other students of this new area, call "electronic publishing" is loaded with land mines.

For the purpose of our discussion today electronic publishing is defined as "the delivery of instructional material through electronic devices in a format familiar to student and faculty". As in any new technology this nomenclature may change. Other identities currently suggested are "coursework" and "academic computing."

Because most of these new materials also integrate other new electronic technologies I feel "electronic publishing" is a more accurate descriptor.

Since I feel the computer will be a common denominator to most electronic publishing techniques in education, I think it is important to summarize the history of the use of the computer in instruction.

HISTORY

While computers were used in many university courses as early as the late 1950s and 1960s, the emphasis at that time was teaching about the computer, i.e., programming languages; configurations and some limited applications. Introduction of the use of the computer in courses such as accounting, engineering and mathematics soon followed since the computer was establishing itself as an economical and useful tool in these functions.

However, the concept of extending the use of the computer directly to the classroom for on-demand, conversational, interactive usage was a product on the 1964-1966 era. Led by the National Science Foundation project at Dartmouth College and Project MAC at MIT, awareness of computer power; ease of use; and some limited programs in classroom teaching began to spread. As in most other pioneer developments a small group of people, who believed in the future of this new-found teaching device, invested not only unlimited hours but millions of dollars in extending this utility concept to the educational field.

These developments took many forms.

(a) Computer service companies installed communication networks to facilitate educational institutions "dialing in" to a computer;

(b) Hardware vendors introduced computers designed specifically for "time-sharing";

(c) Easy to learn computer languages such as BASIC; FOCAL; TELCOMP; QUIKTRAN; and many others emerged with strong arguments for the value of each;

(d) Portable terminals were developed; and finally

(e) Various funding sources such as the National Science Foundation provided grants to evaluate and promote this new tool.

Competition to deliver services and equipment to this market was very intense for such a fledgling area. In the period from 1968 to 1970 a typical high school or college in New Jersey, for example, could have acquired "on-line" computer power from at least ten different sources. Competition, on the other hand, was very geographically concentrated in the Northeast; California and in some random areas where instructional computer pioneers set up camp to offer their wares. Many areas of the country had no resources available.

It is important to recognize that during this period since the early 1960s, of course, computer usage for educational administration programs continues. In some districts these administrative computers are also used for orienting students in a business or technical programming language such as COBOL or FORTRAN and some faculty members have their students writing programs for a course in business or math. At the same time very thorough curriculums have been developed for students bound for a career in the computer field. In these uses the student usually brings his program to the computer center on some predetermined schedule.

It is impossible to fully document the fascinating and rapidly changing history of educational time-sharing, but there are many fine articles and texts describing this history in detail.

The concentration of this paper, however, is directed at the role of the computer in the instructional process as it applies to basic disciplines such as mathematics, science, language, arts, social science and business.

NEED FOR CLASSROOM MATERIALS

Conspicuous by its absence during the growth of time-sharing in the classroom was the availability of supportive text materials - particularly student and teacher manuals; curriculum objectives; and test programs. Studies by the Conference Board of the Mathematical Sciences (1) and HumRRO (2) have highlighted not only the lack of such material, but emphasized that its continual absence would inhibit an efficient or educationally sound use of the computer in the classroom. On the other hand, the emergence of these materials will stimulate the growth of the computer in instruction far beyond expectation.

It is important to examine at this point a framework of how the computer is used in instruction today (see Exhibit A). I have grouped this usage into five major headings:

(a) Computer managed instruction;

(b) Data base retrieval;

(c) Computer assisted instruction;

(d) Computer literacy;

(e) Multi-media management;

Let me define my categories:

(a) Computer Managed Instruction: This category includes those areas of instruction which use the computer in a more traditional format by keeping track of student progress; scoring student responses and branching to a prescribed lesson; comparing class and school building performance; and other record keeping and report generating duties usually required of the classroom teacher. The computer does not teach the student by showing the correct answer; proper formula; correct spelling, etc. Examples of computer managed instruction are shown in Exhibits B and C.

(b) Data Base Retrieval: This area of computer usage is designed to provide the student or teacher with a vast array of information which can be searched through key words or number coded characteristics. One area, for example, would be to provide the student doing research on a famous author with all of the references available in the school library; another to provide a teacher with all of the films in the school media center which discusses a particular topic; another to provide the student and guidance counselor with all of the characteristics on post secondary education and occupational titles. This area will also experience further growth in the distribution of data bases providing compatible reference data on certain tests by grade level; age; and other criteria necessary to measure school performance. A data base retrieval example is shown in Exhibit D.

(c) Computer Assisted Instruction: This area utilizes the computer to react in a purely teaching mode to the students input. These programs can be constructed by the teacher to cover a particular part of the course such as Hewlett-Packard's Instructional Dialogue Facility; or prewritten by an author in basic courses such as chemistry where most responses can be predicted. An example of this application is shown in Exhibit E.

(d) Computer literacy: This area is well defined by the previously mentioned study done by the Committee on Computer Education of the Conference Board of the Mathematical Science. Their definition states that a computer literacy course should:

"(a) Give the student enough understanding about the way the computer works to allow him to understand what computers can and cannot do. Wherever possible, this should involve at least a minimum of direct interaction with a computer, primarily (at this level) through the use of appropriately pre-programmed application packages.

(b) Include a wide sampling of the ways in which computers are used in our society, with non-numeric as well as numeric applications. The impact of these various uses on the individual should be made clear.

(c) Introduce the notion of an algorithm, and its representation by flow charts; where time allows and as equipment becomes available, discuss the manner in which algorithms are represented by programs and the way in which programs are executed by machines". (1)

Many of the computer programs stored for instructional purposes today are designed to familiarize the student with how the computer may be used in a particular subject. These are model programs providing the student with some basis to write his own. (See Exhibit F)

(e) Multi-Media Management Index: This application is relatively new, but growing quickly as new multi-media techniques emerge. In this case the computer "script" with the student can make reference to supporting "off-line" materials such as video lectures on the topic the student is having a problem with in his computer dialogue. These off-line devices can be cassettes, film strips, sound responders, television sets or a combination of these elements. The computer terminal can be designed to actually actuate these devices. These same devices may also be used to retain certain output data and replayed for the teacher who wants to scan the students interaction with the system.

Let me acknowledge at this point my five groupings and definitions represent my own experience and analysis. The field we are discussing today is moving so quickly that there had been little formal research or agreement on these areas. My purpose here is to provide one structure which hopefully is convenient for education computer center managers to use in planning for their own department and working with school faculty, administrators and school boards.

THE EMERGENCE OF TEXT MATERIALS

Many factors have contributed to the establishment of this important area:

(a) Some classroom teachers prepared materials on a local level for use within their school district maintaining the interest level of the computer in instruction.

(b) Two very significant projects funded by NSF - Huntington (I and II) and Project Solo not only "exported" their materials to interested districts, but delivered a framework, or example, for other authors to follow.

(c) Costs of instructional computing started to decline as volume production was introduced.

(d) The number of "computer literate" faculty at the secondary school has increased significantly, and sufficient access and time to develop programs has been made available in many school districts.

(e) The number of new faculty who have been exposed to computing in the classroom not only in college, but in their high school days, is increasing rapidly.

(f) The number of sound instructional programs appearing in new areas such as chemistry and foreign languages has stimulated teachers to evaluate the use of these materials.

(g) Finally, and perhaps the most important factor, educational publishers have recognized not only an opportunity in this expanding market, but find some pressure to defend their text adoptions by providing computer related materials.

The exact penetration of classroom computing at the K-12 level is a very difficult figure, but we estimate that in the fall of 1974 over 1,000 instructional computer systems were providing service at the K-12 level for approximately 2,000 on-line access ports. These instructional computers range from providing one to over 60 simultaneous ports. In New Jersey, for example, we estimate that there are about 200 on-line ports used daily.

MANAGING THE INNOVATION

As I look at this continuing maturity and growth of the computer in instruction from my many visits and experiences over the past few years, one important element is of key concern. To insure economical and educationally sound growth there is a serious need for clearly defined authority and responsibility for its management. As I stated at the beginning of this paper I think the educational computer center manager will emerge as the individual whom most school districts will look to for this leadership and direction.

There are many reasons for this opinion:

(a) Understanding of computer technology, particularly the interface of peripherals, will be a critical skill needed in this function. For example, new student terminals using voice response and home television are being announced and field tested.

(b) The traditional batch processing records management function is already using remote data collection devices and printers. It is a small step to capture this data in the teaching process itself using mark sense readers; cassettes; or time sharing terminals directly.

(c) The managing function must be independent of any one discipline to insure equal sharing of the computer facility among buildings; grade levels; and subjects. Already many instructional computer facilities are dominated by usage for "college bound" curriculums.

(d) The important ingredient, the computer software, or program, to support the instructional program must be evaluated by computer personnel to assure its operation on their equipment. This decision cannot be left to faculty.

(e) As more publishers and hardware vendors introduce new products the school district must provide a clearinghouse as an evaluation process for the various sales forces calling on the school district.

(f) School district budgets will require a broader management role for its current staff as it faces criticism of growing overhead ratios. This factor will make it difficult to have a separate manager for this area.

(g) In those districts apparently succeeding in integrating electronic publishing the educational computer center manager has taken charge. In many cases he is stimulating orientation workshops in instructional computing entrenching his identity with this new area.

In closing my comments I would like to describe some new objectives which the educational computer center manager must meet to insure his success in this new role:

(a) He must develop an acceptance by fellow faculty members that he understands the "realities" of the classroom - that innovation is not easy particularly with new, sophisticated hardware.

(b) He must insure that electronic and computer terminology should be avoided in introducing new programs to the students and faculty. Make it simple and clear.

(c) He should insure that the faculty understands the purpose of these devices is to enhance the teaching process and not to replace it.

(d) He must understand that not all faculty or students will utilize new technologies immediately.

(e) He must insure that decisions have been made and techniques installed to insure privacy of student and faculty information in these new systems.

(f) He must understand that if there is limited time for teacher training many misunderstandings will arise.

(g) He should establish a continuing evaluating method of new programs through his own professional associations such as NAUCAL.

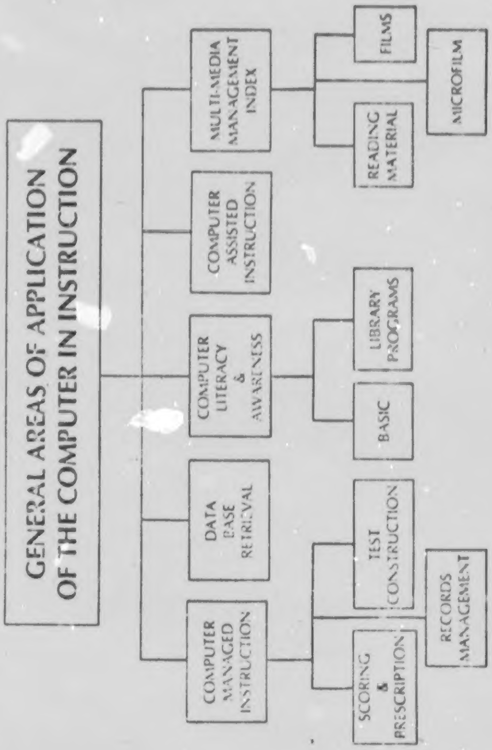
Most of us here today are aware of the rate of change in our industry. It will be a major failure on our parts unless we can translate electronic publishing into useful purposes. In our case this purpose must be to enhance the teaching process by extending these techniques for better, faster and more economical instructional methods. I think it will be your job to make it happen.

REFERENCES

(1) Recommendations Regarding Computers in High School Education; Conference Board of the Mathematical Science Committee on Computer Education; 2100 Pennsylvania Avenue, N.W.; Suite 934; Washington, D.C. 2-037; 1972

(2) Further information may be obtained by writing to Dr. Robert Seidel; HumRRO; 300 North Washington Street; Alexandria, Virginia 22314

Exhibit A


GENERAL AREAS OF APPLICATION OF THE COMPUTER IN INSTRUCTION
MULTI-MEDIA MANAGEMENT INDEX
FILMS
MICROFILM
READING MATERIAL
COMPUTER ASSISTED INSTRUCTION
COMPUTER LITERACY & AWARENESS
LIBRARY PROGRAMS
BASIC
DATA BASE RETRIEVAL
COMPUTER MANAGED INSTRUCTION
TEST CONSTRUCTION
RECORDS MANAGEMENT
SCORING & PRESCRIPTION

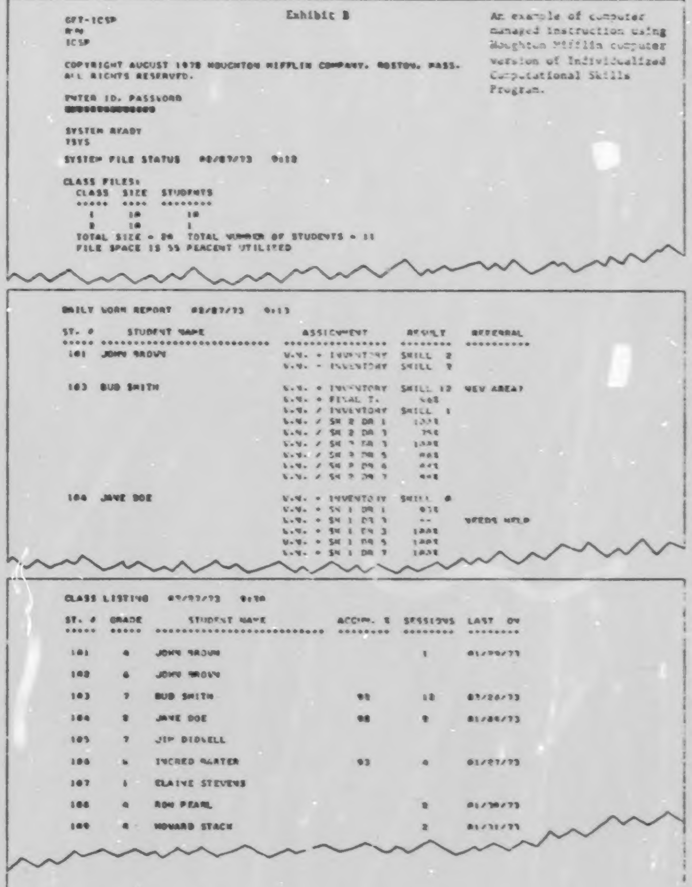


Exhibit B

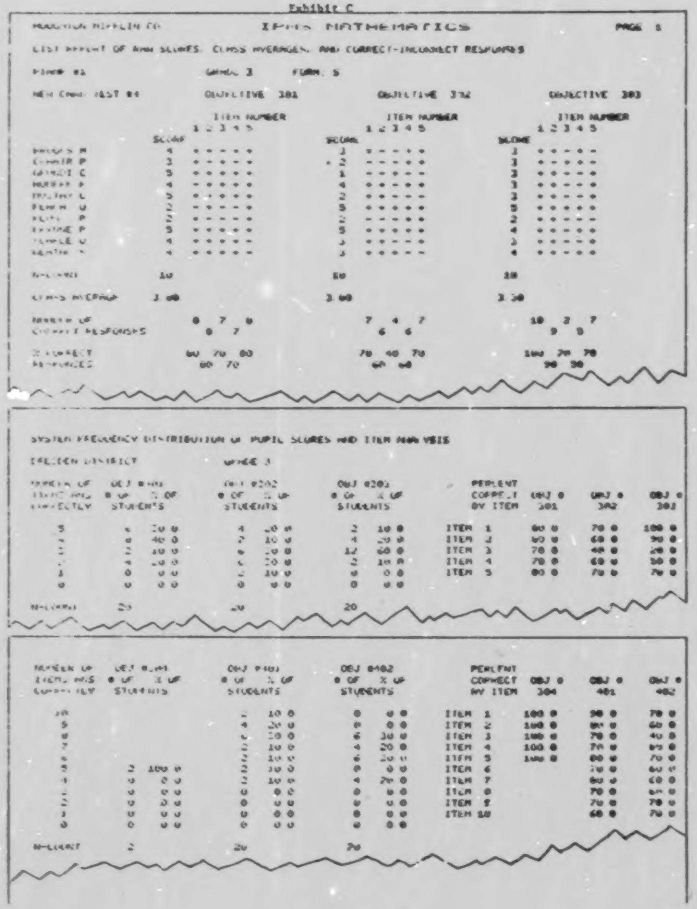
An example of computer managed instruction using Houghton Mifflin computer version of Individualized Computational Skills Program.

```
GFT-ICSP
R-N
ICSP

COPYRIGHT AUGUST 1970 HOUGHTON MIFFLIN COMPANY, BOSTON, MASS.
ALL RIGHTS RESERVED.

ENTER ID, PASSWORD

SYSTEM READY
TSYS

SYSTEM FILE STATUS   02/07/73   9:10

CLASS FILES:
  CLASS  SIZE  STUDENTS
  *****  ****  ********
    1     10      10
    2     10       1
  TOTAL SIZE = 20  TOTAL NUMBER OF STUDENTS = 11
  FILE SPACE IS 55 PERCENT UTILIZED
```

```
DAILY WORK REPORT   02/07/73   9:13

ST. #     STUDENT NAME                ASSIGNMENT        RESULT     REFERRAL
***** ************************** **************** ********  **********
 101  JOHN BROWN                  W.W. + INVENTORY  SKILL  2
                                  W.W. - INVENTORY  SKILL  9

 103  BUD SMITH                   W.W. + INVENTORY  SKILL 12  NEW AREA?
                                  W.W. + FINAL T.      56%
                                  W.W. / INVENTORY  SKILL  1
                                  W.W. / SK 2 DR 1    100%
                                  W.W. / SK 2 DR 3     75%
                                  W.W. / SK 2 DR 3    100%
                                  W.W. / SK 2 DR 5     80%
                                  W.W. / SK 2 DR 6     44%
                                  W.W. / SK 2 DR 7     44%

 104  JANE DOE                    W.W. + INVENTORY  SKILL  8
                                  W.W. + SK 1 DR 1     91%
                                  W.W. + SK 1 DR 3     --      NEEDS HELP
                                  W.W. + SK 1 DR 3    100%
                                  W.W. + SK 1 DR 5    100%
                                  W.W. + SK 1 DR 7    100%
```

```
CLASS LISTING   02/22/73   9:30

ST. #  GRADE      STUDENT NAME              ACCUM. %  SESSIONS  LAST  ON
*****  *****  **************************  ********  ********  ********

 101     4    JOHN BROWN                               1      01/29/73

 102     6    JOHN BROWN

 103     7    BUD SMITH                      95       18      02/26/73

 104     8    JANE DOE                       98        9      01/08/73

 105     7    JIM DIDWELL

 106     6    INGRED CARTER                  93        4      01/27/73

 107     1    ELAINE STEVENS

 108     4    RON PEARL                                2      01/30/73

 109     4    HOWARD STACK                             2      01/31/73
```

Exhibit C

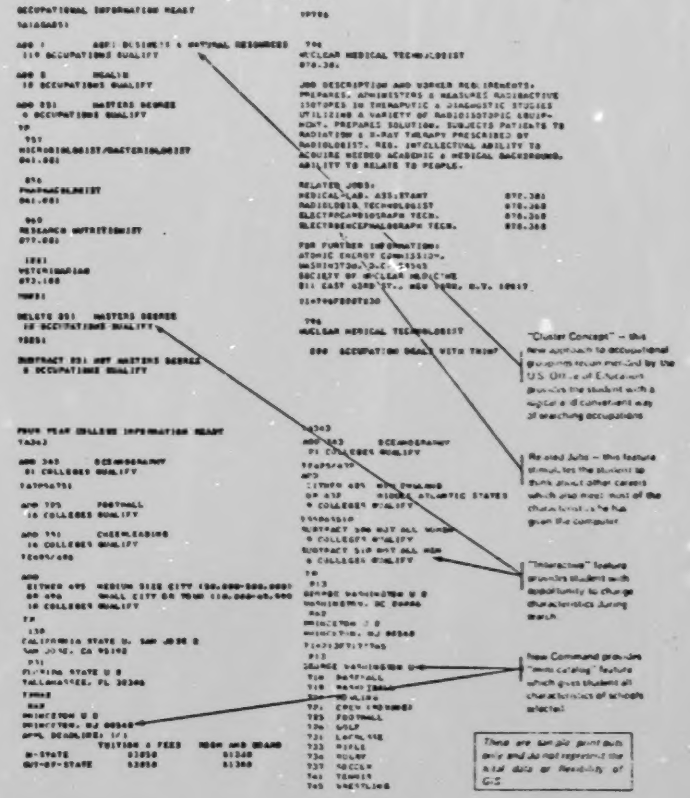


```
HOUGHTON MIFFLIN CO.           ICSP  MATHEMATICS                          PAGE 1

LIST REPORT OF RAW SCORES, CLASS AVERAGES, AND CORRECT-INCORRECT RESPONSES

PIONE #1                  GRADE 3      FORM: 5

NEW CONN TEST #4          OBJECTIVE  301            OBJECTIVE  302            OBJECTIVE  303

                             ITEM NUMBER               ITEM NUMBER               ITEM NUMBER
                             1 2 3 4 5                 1 2 3 4 5                 1 2 3 4 5
                     SCORE                     SCORE                     SCORE
BROOKS M               4     + + - + +           3     + - - + +           3     + + - + -
CHAMIER P              3     + - + - +         * 2     - - - + +           3     + - + + -
[illegible] C          5     + + + + +           1     - - + - -           3     + + - - +
MURPHY P               4     - + + + +           4     + + - + +           3     + + - - +
[illegible] L          5     + + + + +           2     + + - - -           3     + + - - +
PERKIN U               2     - + - - +           5     + + + + +           5     + + + + +
[illegible] P          2     + - - - +           2     - - - + +           2     + + - - -
[illegible] P          5     + + + + +           5     + + + + +           4     + + - + +
TEMPLE U               4     + + + + -           3     + + - - +           3     + + - - +
[illegible]            4     + + + + -           3     + + + - -           4     + + - + +

N-COUNT               10                        10                        10

CLASS AVERAGE        3.80                      3.00                      3.30

NUMBER OF                    8   7   8                 7   4   7                10   2   7
CORRECT RESPONSES              8   7                     6   6                     9   9

% CORRECT                   80  70  80                70  40  70               100  20  70
RESPONSES                     80  70                    60  60                    90  90
```

```
SYSTEM FREQUENCY DISTRIBUTION OF PUPIL SCORES AND ITEM ANALYSIS

[illegible] DISTRICT          GRADE 3

NUMBER OF    OBJ #0301      OBJ #0302      OBJ #0303         PERCENT
ITEMS ANS.  # OF   % OF    # OF   % OF    # OF   % OF       CORRECT   OBJ #   OBJ #   OBJ #
CORRECTLY    STUDENTS       STUDENTS       STUDENTS          BY ITEM    301     302     303

    5          6   30.0       4   20.0       2   10.0       ITEM  1    80.0    70.0   100.0
    4          8   40.0       2   10.0       4   20.0       ITEM  2    80.0    60.0    90.0
    3          2   10.0       6   30.0      12   60.0       ITEM  3    70.0    40.0    20.0
    2          4   20.0       6   30.0       2   10.0       ITEM  4    70.0    60.0    50.0
    1          0    0.0       2   10.0       0    0.0       ITEM  5    80.0    70.0    70.0
    0          0    0.0       0    0.0       0    0.0

N-COUNT       20             20             20
```

```
NUMBER OF    OBJ #0304      OBJ #0401      OBJ #0402         PERCENT
ITEMS ANS.  # OF   % OF    # OF   % OF    # OF   % OF       CORRECT   OBJ #   OBJ #   OBJ #
CORRECTLY    STUDENTS       STUDENTS       STUDENTS          BY ITEM    304     401     402

   10                         2   10.0       0    0.0       ITEM  1   100.0    90.0    70.0
    9                         4   20.0       0    0.0       ITEM  2   100.0    80.0    60.0
    8                         6   30.0       6   30.0       ITEM  3   100.0    70.0    40.0
    7                         2   10.0       4   20.0       ITEM  4   100.0    70.0    80.0
    6                         2   10.0       6   30.0       ITEM  5   100.0    80.0    70.0
    5          2  100.0       2   10.0       0    0.0       ITEM  6            70.0    60.0
    4          0    0.0       2   10.0       4   20.0       ITEM  7            80.0    60.0
    3          0    0.0       0    0.0       0    0.0       ITEM  8            70.0    80.0
    2          0    0.0       0    0.0       0    0.0       ITEM  9            70.0    70.0
    1          0    0.0       0    0.0       0    0.0       ITEM 10            60.0    70.0
    0          0    0.0       0    0.0       0    0.0

N-COUNT        2             20             20
```

Exhibit D



```
OCCUPATIONAL INFORMATION READY
[illegible]

AND 1        AGRI BUSINESS & NATURAL RESOURCES
 110 OCCUPATIONS QUALIFY

AND 5        HEALTH
 18 OCCUPATIONS QUALIFY

AND 331      MASTERS DEGREE
 6 OCCUPATIONS QUALIFY
?P
 757
MICROBIOLOGIST/BACTERIOLOGIST
041.081

 8?6
PHARMACOLOGIST
041.081

 960
RESEARCH NUTRITIONIST
077.081

 1881
VETERINARIAN
073.108

[illegible]

DELETE 331    MASTERS DEGREE
 18 OCCUPATIONS QUALIFY
[illegible]

SUBTRACT 331 NOT MASTERS DEGREE
 8 OCCUPATIONS QUALIFY
```

```
[illegible]

 796
NUCLEAR MEDICAL TECHNOLOGIST
078.381

JOB DESCRIPTION AND WORKER REQUIREMENTS:
PREPARES, ADMINISTERS & MEASURES RADIOACTIVE
ISOTOPES IN THERAPUTIC & DIAGNOSTIC STUDIES
UTILIZING A VARIETY OF RADIOISOTOPIC EQUIP-
MENT. PREPARES SOLUTION. SUBJECTS PATIENTS TO
RADIATION & X-RAY THERAPY PRESCRIBED BY
RADIOLOGIST. REQ. INTELLECTUAL ABILITY TO
ACQUIRE NEEDED ACADEMIC & MEDICAL BACKGROUND,
ABILITY TO RELATE TO PEOPLE.

RELATED JOBS:
MEDICAL-LAB. ASSISTANT                      079.381
RADIOLOGIC TECHNOLOGIST                     078.368
ELECTROCARDIOGRAPH TECH.                    078.368
ELECTROENCEPHALOGRAPH TECH.                 078.368

FOR FURTHER INFORMATION:
ATOMIC ENERGY COMMISSION,
WASHINGTON, D.C. 20545
SOCIETY OF NUCLEAR MEDICINE
211 EAST 43RD ST., NEW YORK, N.Y. 10017

[illegible]

 796
NUCLEAR MEDICAL TECHNOLOGIST

 000  OCCUPATION DEALS WITH THIS?
```

```
FOUR YEAR COLLEGE INFORMATION READY
[illegible]

AND 343      OCEANOGRAPHY
 91 COLLEGES QUALIFY
[illegible]

AND 705      FOOTBALL
 16 COLLEGES QUALIFY

AND 791      CHEERLEADING
 16 COLLEGES QUALIFY
[illegible]

AND
 EITHER 495  MEDIUM SIZE CITY (50,000-500,000)
 OR 496      SMALL CITY OR TOWN (10,000-50,000)
 10 COLLEGES QUALIFY
?P
 130
CALIFORNIA STATE U. SAN JOSE 2
SAN JOSE, CA 95192
 231
FLORIDA STATE U 2
TALLAHASSEE, FL 32306
[illegible]
 862
PRINCETON U 2
PRINCETON, NJ 08540
APPL DEADLINE: 1/1
               TUITION & FEES   ROOM AND BOARD
 IN-STATE          $3850            $1340
OUT-OF-STATE       $3850            $1340
```

```
[illegible]
AND 343      OCEANOGRAPHY
 91 COLLEGES QUALIFY
[illegible]
AND
 EITHER 425  NEW ENGLAND
 OR 432      MIDDLE ATLANTIC STATES
 9 COLLEGES QUALIFY
[illegible]
SUBTRACT 500 NOT ALL WOMEN
 9 COLLEGES QUALIFY
SUBTRACT 510 NOT ALL MEN
 6 COLLEGES QUALIFY
?P
 213
GEORGE WASHINGTON U 2
WASHINGTON, DC 20006
 862
PRINCETON U 2
PRINCETON, NJ 08540
[illegible]
 213
GEORGE WASHINGTON U 2
 710  BASEBALL
 718  BASKETBALL
 720  BOWLING
 721  CREW (ROWING)
 725  FOOTBALL
 726  GOLF
 731  LACROSSE
 733  RIFLE
 736  RUGBY
 737  SOCCER
 741  TENNIS
 745  WRESTLING
```

"Cluster Concept" – this new approach to occupational groupings recommended by the U.S. Office of Education provides the student with a logical and convenient way of searching occupations

Related Jobs – this feature stimulates the student to think about other careers which also meet most of the characteristics he has given the computer

"Interactive" feature provides student with opportunity to change characteristics during search

New Command provides "mini catalog" feature which gives student all characteristics of schools selected

> *These are sample print outs only and do not represent the total data or flexibility of GIS*

Printouts from Time Share Corporation's Guidance Information System.

Exhibit E

```
WELCOME TO THE CHEMISTRY PROGRAM LIBRARY

WHAT IS YOUR NAME?DOUG
THANK YOU, DOUG

WHICH OF THE PROGRAMS LISTED IN YOUR MANUAL
WOULD YOU LIKE TO TAKE TODAY?
TYPE IN THE NAME OF YOUR CHOICE?LIST

LOG1    SCIV01  SCIV02  MET1.   MET2    TEMCO   DECAL   GRAPH   PPTEL
ATOM    GAS1    GAS2    IDEAL   FORM1   NORIN   MOTER   MOLE1   MOLE2
SOL1    SOL2    SOL3    COPRO   NOANE   QUANE   NOGEN   PHOMA   PERMA
BLEAP   PERCH   GRAVE   CALEX   GRAPE   OLONE   AROMA   OPTRO   TITRT
PH

TYPE IN THE NAME OF YOUR CHOICE?EQBAL
          BALANCING EQUATIONS

ENTER THE COEFFICIENTS (INCLUDING 1) THAT ARE NEEDED
TO BALANCE THE FOLLOWING EQUATIONS.  PLEASE NOTE THAT
THE COEFFICIENTS MUST BE SEPARATED BY DASHES (-). YOU
MAY SKIP PROBLEMS BY ENTERING SKIP, AND YOU MAY STOP
AT ANYTIME BY ENTERING STOP. KEEP SMILING, FRIEND.

C2H6 + O2 = CO2 + H2O
?2-7-4-6

GEE-WOW, DOUG

AL + O2 = AL2O3
?1-3-1
  NO, THE CORRECT COEFFICIENTS ARE 4-3-2

ZNS + O2 = ZNO + SO2
?2-3-2-2

GEE-WOW, DOUG

CL2 + NAI = NACL + I2
?1-2-2-1

GEE-WOW, DOUG

NANO3 = NANO2 + O2
?2-2-1

YIPPEE, DOUG

H3PO4 + KOH = K3PO4 + H2O
?1-3-1-3

GEE-WOW, DOUG

NAOH + H2SO4 = NA2SO4 + H2O
?STOP

YOU CORRECTLY ANSWERED 5 OF 6
FOR A SCORE OF  83.3333 PERCENT.
THANKS FOR YOUR TIME AND SMILES, DOUG


WOULD YOU LIKE TO TRY ANOTHER MODULE, DOUG?NO


OK, TIME FOR ANOTHER STUDENT, OR TO SAY GOODBYE
```

Printout from Time Share Corporation's CHEMCOMP Program.

Exhibit F

```
POLUT

                 WATER POLLUTION STUDY

INSTRUCTIONS (1=YES, 0=NO)?1


IN THIS STUDY YOU CAN SPECIFY THE FOLLOWING CHARACTERISTICS:

A. THE KIND OF BODY OF WATER:
   1. LARGE POND
   2. LARGE LAKE
   3. SLOW-MOVING RIVER
   4. FAST-MOVING RIVER

B. THE WATER TEMPERATURE IN DEGREES FAHRENHEIT:

C. THE KIND OF WASTE DUMPED INTO THE WATER:
   1. INDUSTRIAL
   2. SEWAGE

D. THE RATE OF DUMPING OF WASTE, IN PARTS PER MILLION (PPM)/DAY.

E. THE TYPE OF TREATMENT OF THE WASTE:
   0. NONE
   1. PRIMARY (SEDIMENTATION OR PASSAGE THROUGH FINE
               SCREENS TO REMOVE GROSS SOLIDS)
   2. SECONDARY (SAND FILTERS OR THE ACTIVATED SLUDGE
                 METHOD TO REMOVE DISSOLVED AND COLLOIDAL
                 ORGANIC MATTER)
************

BODY OF WATER?3
WATER TEMPERATURE? 70
KIND OF WASTE?1
DUMPING RATE?8
TYPE OF TREATMENT?0


DO YOU WANT: A GRAPH(1), A TABLE(2), OR BOTH(3)?1

AFTER DAY 3 THE FISH BEGIN TO DIE, BECAUSE
THE OXYGEN CONTENT OF THE WATER DROPPED BELOW 5 PPM.


        0...OXYGEN-SCALE....5...OXYGEN-SCALE...10...OXYGEN-SCALE...15
        0..WASTE.10..SCALE.20..WASTE.30..SCALE.40..WASTE.50..SCALE.60
DAY     I---------I---------I---------I---------I---------I---------I
 0      I  W                          O
 1      I         W                O
 2      I              W        O
 3      I                   WO
 4      I                 O    W
 5      I               O         W
 6      I             O             W
 7      I            O               W
 8      I           O                 W
 9      I          O                   W
 10     I         O                     W
 11     I         O                      W
 12     I         O                      W
 13     I        O                       W
 14     I        O                        W
 15     I        O                        W
 16     I        O                        W
 17     I        O                        W
 18     I        O                        W

THE WASTE CONTENT AND OXYGEN CONTENT WILL REMAIN AT
THESE LEVELS UNTIL ONE OF THE VARIABLES CHANGES.

ANOTHER RUN (1=YES, 0=NO)?1
```

Source: Huntington Project

COMPUTER READINGS

Edward O. Joslin

I have been asked to briefly discuss some of my experiences in preparing the ten books of readings I have compiled. I took that invitation to mean, describe the steps involved in compiling a book of readings. However, before I describe the "how to" aspect, I feel I should say something about why I started doing books of readings in the first place.

What Purpose Is Served By Readings

In some of the computer courses I was teaching in 1967 and 1968, I was assigning chapters for the students to read and then I was lecturing to the students on the material from those same chapters. This repetition, while of some educational value, tended to make those courses rather boring and certainly was not making effective use of the students' time nor efforts.

To help broaden the courses and to introduce an element of currentness in them, I began assigning outside articles for the students to read. This not only put a strain on the students' time and on the library facility, but also on me as I constantly had to find new articles for them to review.

Next, in order to minimize the time being wasted by the students in searching out the assigned articles, and to spare the library all the problems associated with many students seeking out the same articles at the same time, I put together a group of twenty or thirty pertinent articles and had them reproduced for the students' use. This reproduction of the articles overcame the students' and library's problems, but created new ones for me. The reproduction cost of twenty or thirty articles for twenty or thirty students was considerable, in fact so considerable it was never repeated. Also it was illegal.

The prohibitive cost of reproducing the articles, caused me to consider compiling a book of computer readings. I already had a good start on compiling a set of readings, but having already gone through the very long and time-consuming process of having had a book published (Computer Selection) I was reluctant to go that route again. Especially, with a book of readings, which was necessitated in part by a need to present current information which was missing in the prime textbook, because they had to go through the long and time consuming publication process. Therefore, I started searching for a publisher who would talk in terms of a six month or less publication time. Believe me there are few. Finally, I got together with College Readings, Inc. (P. O. Box 2323, Arlington, Virginia 22202), my current publisher, and that problem was solved. It now takes less than three months from the time I submit the complete set of material until the finished text is printed and ready for distribution.

Having found a way around the long delays involved in most book publications, I once again started to concentrate on compiling that first book of readings - the steps of which I will be discussing in just a moment.

But first, let me skip forward a few years and discuss some more of my own experience to show additional reasons why someone may be interested in using a book of computer readings. The year is now 1969, I had already compiled my first two books of readings (An Introduction to Computer Systems and Management and Computer Systems) and I was now asked to teach a programming course. I was appalled by what I found. The programming course that I was to teach was the only software course being given the students. The course was designed to teach the students how to program a given machine, and thus to give them a familiarity with programming concepts. But in no way did it even begin to make the students aware of the bigger, more important, world of computer software, i.e., compilers, operating systems, application programs, etc. Therefore, I started compiling a book on Software for Computer Systems. That text is an example of a book which was compiled to serve a very different purpose than the assigned textbook - namely to go into areas not even intended to be covered by the prime text. But yet it was a supplement to the prime text.

Another example of the compilation and use of a book of readings, is illustrated by my text

Analysis, Design and Selection of Computer Systems. This text was designed in part to serve as a replacement for my Computer Selection book, rather than go through the time consuming process of revising it. I rewrote some chapters of that text as articles and, along with other articles, put them into my compilation on the subject. Thus, that compilation is being used in many courses as the only text - if the emphasis is on selection. In this case the compilation is the only text available on the subject. The vehicle of a book of readings was used rather than a normal authored text because it could be published much more quickly.

Thus, from my own experience, I have attempted to show three uses of books of readings: 1) to supplement and bring currentness to an existing course; 2) to broaden (by bringing a full new area of coverage - as in the software case); and 3) to use as a prime text in areas not covered, or not covered as well, by other texts.

Having now seen how and why compilations might be used, let me return to telling you something of what is involved in compiling a book of readings.

Steps In Compiling A Readings Book

First, if you are compiling the book as a means of making a small fortune, forget it. Take up collecting empty soda bottles or picking grapes, you will stand a better chance of making your fortune.

Second, now that you do not need to worry about sharing your fortune, seek out a co-compiler. You will need him (or her) to broaden your scope and to share the work.

Third, determine the needs for the book you have in mind. In my case, since I was into readings books already, I did my doctorial dissertation of "Computer Education: College Offerings and Industry's Requirements". In so doing I surveyed more than 2000 colleges and universities to determine what data processing courses they were offering. I also surveyed more than 1500 companies to determine what they were seeking in the way of data processing education. As a result of those surveys an interesting fact developed. Industry was very interested in having its prospective employees familiar with various applications of computer systems, but the colleges were not offering many courses which did that. One of the reasons the course was not being offered by the colleges was that there was no suitable text. Needless to say, I attempted to take care of that situation with a text entitled, Applications of Computer Systems. That text was compiled as a supplemental text for use in an introductory course.

Fourth, a determination needs to be made as to the intended use of the text. Is it primarily intended as a supplemental text or as a primary text?

Fifth, some attempt must be made to determine the market potential for the text. You may feel there is a crying need for a text on "Data Security". But if the book were prepared as a textbook, would there be sufficient demand to warrant its publication.

Sixth, outline the subject areas that should be covered. One thing to remember in any such outline, is to make it broader than your own needs would dictate. Not everybody will see the needs as you do, some areas you may consider trivial will be sought by others, while they are skipping areas you consider critical. A book, particularly a book of readings, needs to be broad in coverage.

Seventh, seek out a publisher. I recommend College Readings Inc., who specializes in readings books, but there are others. Check to see how long it will take to publish the book once you supply all the materials.

Eighth, start selecting articles that cover the subject areas you want covered. In my selection of articles I stress "currentness" right along with "content". If the article is not current, but it has good content, I have a tendency to assume that its content will be picked up in some recently authored text, and therefore I will continue searching for current articles with good content.

Ninth, search out some of the less common publications.

Tenth, start preparing, or having prepared, some special articles for the areas you want covered, but for which you could not find any good or current articles. The more specialized your subject areas, the more likely you'll have to author some of the articles. These authored articles tend to take a long time to produce and clean up.

Eleventh, start obtaining permission to use the articles you have selected, and do not count on any article until you have secured the necessary permission to use it. Permission will frequently be denied because the author is using the material in a book he is preparing himself. Sometimes the publisher or author will ask a fee so outrageous that you can not afford to use the article. (I have had several requests for $500.00 to $1000.00 per article - and I use from forty to fifty articles per book.)

Twelfth, now at a point in time much later than you ever believed possible (it takes a long time to write, or have written, articles and a long time for some permissions to come through, especially when you have to track down some of the authors), take the articles for which you have received permissions and the articles you wrote, or had written, and see what kind of book you are ending up with.

Thirteenth, if the restructured book is still worth publishing, deliver it and the permissions to use all the articles to the publisher.

Fourteenth, stand-by to review the page proofs of the text and use them to prepare an index for the book.

Fifteenth, upon publication of the text, be very humble in accepting any praise you may receive for having compiled a book of readings. It is necessary to maintain the humble attitude because in the majority of the cases in which you received the praise, the next sentence will probably be "But why did you, or did you not, include the articles: ....."; and if that does not make you feel humble, just wait until you receive your first "Congratulations" followed by a "But then it is only a compilation of other people's work".

Books Compiled by Joslin for College Readings Inc.

An Introduction to Computer Systems, 1969, 1972 and 1975.

Management and Computer Systems, 1970 and 1973.

Software for Computer Systems, 1970.

Analysis, Design and Selection of Computer Systems 1971 and 1974.

Applications of Computer Systems, 1974.

Computer Readings for Making It Count, 1974.

USE OF COMPUTERS FOR INSTRUCTION IN THE PHILADELPHIA SCHOOLS

SYLVIA CHARP, DIRECTOR
SCHOOL DISTRICT OF PHILADELPHIA

PANELISTS:

Sylvia Charp, School District of Philadelphia
Henry Altschuler, School District of Philadelphia
Richard D'Orazio, School District of Philadelphia

4

The School District has been involved with the use of computers for instruction for the past ten years. Many programs have been initiated during the period. These programs cover the areas of computer-assisted, computer-managed instruction, and problem-solving. The panelists will discuss their experiences and offer suggestions to other schools considering use of the computer.

OTIS: Information System for Oregon Educators; Instructional Computing for Oregon Students

Jill Van't Roer
Oregon Total Information System

In the mid-sixties Oregon had no large and comprehensive educational data processing system. In 1966 a government planning grant initiated the project which was to become Oregon Total Information System; in May, 1968, OTIS began providing data processing services from its central location in Eugene, Oregon, primarily to schools in the surrounding county (Lane).

Today OTIS has become a successful, completely user funded cooperative of school districts administered by the Lane County Intermediate Education District (IED). The Lane IED Superintendent is Dr. William C. Jones; OTIS' Director is Robert L. Dusenberry. OTIS serves disparate school systems (elementary, secondary, some community colleges) and educational agencies throughout Oregon with administrative and instructional services.

OTIS' systems are designed for its users; its services are utilized efficiently and creatively by school personnel, using terminals at their own buildings, without the necessity of any more data processing knowledge than that which they can get from manuals supplied by OTIS. Clerks and business officers input and update information to their files in the financial accounting, payroll, personnel and inventory systems, and reports are generated from their input. Students are enrolled, graded, and scheduled by OTIS; attendance reports are run from terminal input, and inquiry capabilities make the information in individual files always immediately available. OTIS' on-line Query System allows the user to create his own reports: using a Query manual of statements he can search files and extract pertinent facts and so adapt reports to conform to his specifications. Students can learn BASIC, a programming language, write their own programs, store them and rerun them as often as they wish. Instructional media center personnel can use a terminal transaction to book materials, and the booking is immediately confirmed for the requested date, or if it is not available, the user is informed of the next date it will be.

Through the teleprocessing network schools can access the IBM S/360 model 50 with Ampex add-on fast core, which holds administrative services files, or the two Hewlett-Packard 2000F computers used for instructional services, and can be switched from one machine to another by the GTE IS1101.* The General Education Management System (GEMS), an OTIS software system, is a system of integrated data files with two sets of control tables and a set of generalized processing programs. GEMS makes it possible for each user to define and maintain only the data storage and processing that he requires and to change his definitions as his needs for information change.

OTIS SERVICES - Administrative

Have OTIS' services grown and become increasingly sophisticated because increasing user needs have made it necessary, or has usage of the system grown because new and more comprehensive services have become available? An unanswerable question. Perhaps OTIS' success is due to the great need recognized by Oregon school personnel for a large data processing system, or perhaps OTIS' success is due in large part to its creative staff members, who have developed original software and many modifications to existing software during the establishment and expansion of OTIS' many services.

The emphasis of OTIS' first year of operation (1968-69) was on Student Services. Some software had been developed by Lane Intermediate Education District personnel, some from vendors. The three main working systems were testing, enrollment, and mark reporting. The attendance system, totally developed at OTIS, was in its formative stages, and the completed system was debugged and usable by 1971.

Educational Coordinates' (EC) scheduling system and the Student Scheduling System (3S) are both used at OTIS. In 1968 OTIS gathered data and sent it to EC for input; in 1969 OTIS acquired EC's library of software and could do all the scheduling in-house. A loader program from IBM started the 3S system, which has been modified extensively. OTIS staff developed an arena scheduling system which was implemented in 1971. As scheduling needs have changed so have systems. In the last five years there has been a dramatic move away from structured, standard scheduling, and OTIS has modified its systems to allow for a large variance in scheduling applications.

---

*See Computerworld, July 31, 1974, pages S11-S12, for a more complete discussion on OTIS software modifications to the GTE IS1101.

By the end of 1972 the Student Services system was functional and completely implemented. The systems have since evolved to allow great flexibility in types of output. Student body rosters or mailing labels, for example, can be printed alpha by grade, by teacher, by sex, etc. Labels for tenth grade boys or a list of ninth grade members of the band can be generated easily.

The Query System has been under development since the fall of 1968. The original system, conceived by an IBM analyst, was slow and inflexible, with little choice in sort order. Priority was given to increasing the speed of the system, finding means of developing the most useable sorted output, and allowing the user to specify what sort order he wished. As data becomes large, the sort order is more important; a non-alphabetical list of all boys in a 1000 student school would have been unwieldy and probably unusable. This year, 1974-75, the Query System is dependable, efficient, and is on-line. The user can enter his query via terminal and will get his report back on the terminal the same day, sometimes almost immediately.

A new system at OTIS is a automated record keeping system developed to accomodate Oregon's high school graduation requirements. The State Department of Education has directed that certain requirements be met by students in Oregon schools: In order to graduate from high school a student must prove minimum survival level competencies in three general areas - personal development, social responsibility and vocational education. "Minimum survival level competencies" have been described by an Oregon educator as "what you have to know in order not to be taken too badly or too often." Since each district develops and evaluates its own competency levels, the record keeping system has had to be extremely flexible to allow for large individual user needs. The system offers individual district options on definitions of competencies, multiple-check status of competencies, types of input, and types of reports. This year three districts of varying sizes and problems are piloting the system.

Business Services provided access to a payroll and a fiscal system its first year (1968), and to an inventory system, which was used by only four districts.

Payroll information was input by batch processing at the computer center [now it is completely online; i.e., district clerks input all the information needed from their own terminals], and processing of both payroll and fiscal report runs was set up on demand rather than on a schedule. Processing was slow, and OTIS staff members spent many extra hours keeping the system running smoothly. In the 1974-75 school year 56 districts use the system, and nightly processing runs are scheduled. In 1969, on occasion, one job could take fourteen hours. Now the total run (processing for 56 districts) will take 1/2 to 1 1/2 hours to process.

The Business Services systems have been completely designed by OTIS staff members in conjunction with users. This year services include fiscal, payroll, personnel, inventory and cafeteria inventory accounting systems, which produce over 50 different kinds of reports. Query capabilities are also available.

The newest application is the processing system developed to accomodate Handbook II (revised), an accounting system established by the federal government for school districts. All schools must use the new system by the 1976-77 school year, and the OTIS system is now 90% complete.

In 1972-73 a pilot district used the Handbook II (revised) accounting system, and with modifications OTIS' processing system is now used in seven districts, with thirty expected on in 1975-76. All OTIS Business Services user districts will be ready to use Handbook II (revised) in 1976-77.

OTIS SERVICES - Instructional

In 1969 OTIS instituted an instructional computing system called RAX (Remote Access Computing System). Users could access it for only two hours, on Tuesday and Thursday, from 5-7 p.m., when teleprocessing was down, because it needed to take over the whole IBM 360 system to work. Few terminals used the system, and it was not very successful in terms of convenience, so in 1971 ITF (Inreractive Terminal Facility) was begun. It could be run during the day and offered problem solving and a method of teaching programming to students. However, the ITF system caused problems: It often interfered with teleprocessing and brought the system down, and the turnaround time was extremely slow. There were few canned programs available, and users found the system too limited and undependable.

In 1972 OTIS acquired an HP 2000F computer for instructional use, and a GTE IS1101 to provide interface between the IBM system and the HP. Thus administrative services users can also use the in structional system from the same terminal. The system has 32 ports available for use 24 hours per day, and users pay $3.00 per terminal hour for its use. This fall OTIS has acquired another HP 2000F with option 205 to accomodate increased instructional use. OTIS' instructional computing system serves 42 districts, who used approximately 30,000 terminal hours last year.

Instructional Services currently available include the Computer Related Instruction System (CRIS), the Occupational Information Access System (OIAS) and the OTIS Automated Library System (OALS). CRIS and OIAS use the Hewlett-Packards, OALS uses the IBM GEMS system.

OALS' booking and scheduling system, a service contracted by the Lane IED - Instructional Media Center (IMC), and available to Lane County teachers, students and librarians, schedules over 12,000 discrete print and non-print items for use in schools. The system is on-line and produces operation and usage reports as needed.

The non-print evaluation system is also used in Lane County. Teachers preview and code evaluations of IMC materials in areas of their specialty, and evaluation reports are available for the IED to use in the decision-making process concerning acquisitions. A report can be sent to schools which lists items recommended as useful in different areas, as well as those deemed not useful. Pricing figures and vendors' names are listed.

The book catalog system, used by several county IED's, manipulates input of bibliographic data, subject headings, annotative information and call numbers to make book catalogs, replacing much of the labor of manual card catalog systems.

Literature searches of current educational reports, monographs and journal entries, called ERIC searches, are available to users through the OALS system. OTIS processes nearly 100 searches each month.

CRIS provides instructional computing for all areas of a school's curriculum. CRIS users access over 400 canned programs, such as drill and practice in math, reading and language arts, simulations and games. The system offers problem solving in BASIC for students to learn programming. Computer assisted instruction (CAI) in reading, mathematics and language arts on the elementary level is being piloted this year at Blossum Gulch Elementary School in Coos Bay and plans are underway for a program at Chemawa Indian School.

Software from the Computer Curriculum Corporation (CCC) is used for CAI at the elementary school. Eight terminals at Blossum Gulch are connected to OTIS, and a reading project began this fall. 100 students per day use the system.

CAI is an "exchange of information between computer and student" according to CCC literature. It is aimed at students with a low level of performance; the curriculum for each student "will be at a level appropriate to his achievement in each learning area" so each student has the opportunity to learn at his own level and his own speed. The program provides remedial assistance in the form of progressive drill and practice.

There are advantages to this type of reading and language arts program for remedial students. There is no discipline involved, no teacher disapproval, no necessity to perform for others, and wrong answers won't hurt. The student will have nothing to lose, and it might even be fun.

Chemawa Indian School, an alternative school administered under the Bureau of Indian Affairs, is planning a large program in computer assisted instruction in reading and language arts to begin March 1975. Chemawa is an off-reservation boarding school for dropouts and students expelled from other schools, located in Salem, Oregon, with students from Oregon, Washington, Idaho and Alaska. Counseling and vocational training are emphasized, and students are given a chance to acquire skills they have missed in the formal educational environment.

Chemawa students have not performed well in reading and language arts examinations; according to Chemawa administration, causes for this include lack of time for sufficient individualized instruction and low student motivation caused by past educational experiences. CAI will give individual drill every day, and will be an entirely different educational experience for the students. The project will involve the use of 32 terminals, another HP 2000F at OTIS, and students will use CAI during the day and evening.

OIAS is a computerized vocational counseling program. Through a questionnaire a student answers questions and states preferences, then possible occupations are presented which correspond to the personality profile suggested by the questionnaire results. Detailed information on each occupation is produced upon demand. One can ask for a job abstract, which describes job duties, working conditions, salary range, turnover rates, qualifications for the job, employment prospects and other pertinent facts. Qualitative information is included in this job abstract; for example, the job description for college teacher states that one must have the ability to work with abstract concepts, have self-discipline and good judgment.

Through the INFO files one can also get a bibliography of occupational books, find out how people prepare for the occupation, get a list of courses, degrees and schools for an education or training program, compare the services and costs of schools and get the names of people for personal discussion about the occupations.

The vocational counseling program is jointly administered by OTIS and the Career Information System, an interagency consortium providing vocational information, located at the University of Oregon.

OIAS use is expanding. The Oregon State Correctional Institution and Oregon State Penitentiary now offer the system for their inmates. Lane County data processing personnel are currently working on software to accomodate OIAS on their network. The county organization plans to be able to switch on to the vocational counseling system and use it for jails and social service agencies.

OTIS has grown and changed considerably since 1968. Its charges have lowered from $8.80 per student for complete administrative services to $7.00. The student base of user districts and educational agencies has raised from 65,000 to 150,000, from 28 school districts to 72. OTIS is financially stable and operationally sophisticated, and its services become constantly more varied and complex.

To offer complex services, one must balance them with a sophisticated data base - an education process is continually necessary to train new district or school employees and explain new services, so they can be efficiently and productively used. To facilitate this balance OTIS has a knowledgeable field service staff and a system of data processing coordinators.

Cooperation and communication between users and OTIS are important, to keep misunderstandings down and user operations running smoothly. The data processing coordinators work as liaison between districts and OTIS. Each district's coordinator assists in training of users and communication of district desires and needs. Coordinators meet periodically to discuss problems, new services and procedure.

OTIS' field service staff, each person an expert in a specific service and with a general working knowledge of all services, is a direct communication line between user and OTIS. When a school comes on the OTIS data processing system a representative of each service area shows him how and why the system works. When districts need help, or hire new personnel, field service staff people are ready to smooth the way.

Users from fourteen counties in Oregon are served by OTIS through over 200 terminals. Representatives from county and district educational organizations throughout the state make up OTIS' Advisory Committee, which is consulted on important policy matters and system development.

In terms of upgrading quality and furthering equality of education, OTIS' impact on Oregon school systems has been great. Clerical duties are minimized at school and district level; educators have more time for students, and students have access to instructional computing systems. A district with 200 students can get the same types of reports and information as a large district, with more students and a larger tax base. Only a few districts could afford to provide the services OTIS can provide; through the on-line teleprocessing system all school districts have an equal opportunity to avail themselves of data processing and instructional services.

OTIS HARDWARE

Computers

1 IBM S/360 model 50 with Ampex add-on fast core
1 GTE IS1101 Computer
2 2000F Hewlett-Packard Computers

Input/Output Devices

4 2400 Series Magnetic Tape Units
2 2314 Direct Access Storage Facilities
1 2321 Data Cell Drive, Model 1
1 2540 Card Read Punch
1 1403 Printer, Model N1
3 Double Density Disk Drives

Jill Van't Roer
Oregon Total Information System
354 East 40th
Eugene, Oregon 97405

March, 1975

DEIC

NEW JERSEY'S SYSTEM FOR BRINGING DATA ELEMENTS AND ITEMS UNDER CONTROL

Dr. R. Kay Maloney
Graduate School of Library Service
Rutgers University

The New Jersey State Department of Education (SDE) has a problem -- a data control problem. In 1862, reports from Town Superintendents to the State Superintendent of Public Instruction in New Jersey contained only 26 items. In 1973, an inventory of the data items collected by SDE revealed that the total was in the neighborhood of 26,000 items, an average increase of well over 2,000 new items per decade. If SDE files were to record information from the 600 school districts and 2,400 schools in the state and if this data were retained for a minimum of three years, the volume of data stored would be over 15,000,000 items from school districts and 60,000,000 from local schools, for a total of more than 75,000,000 items per year -- just under a quarter of a billion data items for the three-year period. Furthermore, these figures apply only to data collected from the elementary and secondary schools in New Jersey.

The New Jersey State Department of Education has a second problem -- a legislative problem. During the first half of calendar 1974, SDE was asked to comment on more than 100 pieces of proposed legislation; it was also asked to submit estimates of what the legislation would cost if enacted for at least two succeeding years. Since legislation is proposed by persons from many backgrounds, the bills frequently use words and definitions that would not be considered standard terminology by educators. However, if a bill becomes law, the words or terms therein must be used as stipulated, regardless of whether educators accept the terminology as standard in their field.

The impact of these two problem areas on data collection, processing, and retrieval is enormous. On the one hand, we must deal with storage and control of vast files of statistical data. On the other hand, the data store must be accessible to educators and legislators alike through terminology that communicates the same meaning to both groups of users. It is rather obvious that a system for control of educational data is necessary to resolve these problems.

Whenever large files of information must be stored and manipulated it is reasonable to examine the potential of computer technology to create the files and to access the information. SDE has access to a state-wide computer network that can be used to maintain a central file of educational data. Furthermore, a 1973 survey of New Jersey's 600 public school districts shows that 208 districts, serving nearly 70 percent of the total school population in the state, use some form of automatic data processing equipment to process their records. Well over half of these districts own or lease their equipment.[1]

It is apparent that local educational units are already using automatic data processing in an effort to cope with increased data reporting requirements. Unless SDE coordinates data control by providing a state-wide system, it is likely that independent local systems will proliferate, without concern for compatibility with other systems in operation in the state or nation.

We have established that the New Jersey State Department of Education is responsible for collecting extensive data from New Jersey schools for use at both the federal and state levels. Since the collection of this data originates from a number of departments within SDE, unless a specific effort is made either (1) to coordinate all data gathering through one office, or (2) for each department to confer with every other one before data is collected, unnecessary duplication of information and effort is inevitable. In addition, if no standardized terminology is available for use in collecting data, differing terminology between forms and differing interpretations of what information is to be recorded affect the consistency and therefore the reliability of data actually collected. Not only is the validity of evaluating or comparing such data called into question but the communication of information under such conditions is unreliable.

It would not be sufficient for SDE's purposes merely to "clean up" the data collecting process and its resultant data. It is equally important to improve and to expand the access to the data collected. At present, access to data gathered is limited primarily to those who collect it -- not because of a desire to restrict access to the information, but because of the lack of a central system for storage and retrieval of the data. It is especially difficult for those persons responsible for providing the data in the first place to retrieve data of interest to them. This problem is not unique to New Jersey. A recent issue of ERIC's Interchange indicates the extent of concern for expanded access to educational information:

---

[1] Gaydos, Irvin A. Survey of New Jersey Public School Districts Using Data Processing Equipment-ment. Trenton, N.J., State Department of Education, April 1973.

> The teacher in the rural school must be provided the same retrieval opportunity as the professor of a large university. A superintendent in a district must have the same access to ERIC as the graduate student laboring on his Ph.D. dissertation. A major thrust to provide for these needs must soon surface on a national level -- or a kingdom may someday be lost for lack of a nail.[2]

SDE recognizes the need for an educational data control system: (1) to facilitate efficient storage and expanded retrieval of data collected; (2) to reduce to a minimum the collection of duplicate data; (3) to improve the precision with which data is requested and therefore the reliability of the data itself.

In order to meet these objectives it was decided: (1) to plan a computer-based system for storage and retrieval of educational data, with an on-line, multiple terminal capability both for recording data into the system and for retrieving data from the system; (2) to develop a standardized terminology, comparable to the extent reasonable with professional and national trends, for collection, storage, and retrieval of the data; (3) to provide an indexing structure that will serve as an organizational frame of reference for both storage and retrieval of the actual data.

Thus DEIC -- New Jersey's system for bringing educational Data Elements and Items under Control -- was born.

A project of this scope requires expertise from a variety of disciplines -- education, information science, computer technology. The research component of DEIC reflects these backgrounds. As financial resources for the project became available, SDE approached the Rutgers University Graduate School of Library Service with a research grant proposal that would fund the first phase of development of an indexing structure and a standardized vocabulary. In October 1974, the research team was in full operation with the following primary personnel: Dr. R. Kay Maloney, Project Director, with research experience in subject analysis and computer-based information systems; Kay McGinty, full-time research associate, with experience in social science research; Irvin Gaydos, Assistant Director, Office of Research Information, and liaison to the DEIC Project for SDE; Barbara Smey, candidate for the Ed.D. at Rutgers, serving an educational internship on the project.

Our methodology really follows a progression that evolves naturally when one takes the time to examine fully the objectives of the DEIC Project. First, we must identify what data is actually being collected. Next we must organize that data for storage. Finally, we must index the data so it can be retrieved at a later date.

Identifying the data collected necessitates our securing copies of every form being distributed by each of the departments of SDE that is to be incorporated into the data base. (There are approximately 450 forms, though this is a constantly changing component of the system.) It is then necessary to identify every single request for data on each of these forms. We call this single unit of information a data item. The keyword indexing approach was used to create a working file of the data items appearing on SDE forms. Even from this rough data base we have been able to identify a 25 percent rate of exact duplication of data items. (It should be kept in mind that some duplication is necessary and desirable.) Although we do not have precise statistical data as yet as to the extent of synonymous but not exactly-worded data items, it is our feeling that another 10 percent may fall into this category.

Developing a viable indexing structure that will serve as an organizational framework for the system requires one to take a hard look at exactly what information might be needed in the future and thus must be stored. The data item is not sufficient. Thus far we have isolated six pieces of information for each data item that we consider essential to be stored in the system: (1) a description, in our standardized terminology, of the data item requested; (2) the statistic (the "answer") collected; (3) the geographic and/or school unit giving the data; (4) the time span that the statistic collected represents; (5) identification of the form on which the data item appears; (6) the collecting agency within SDE. Other information necessary for system operation is also included, but is not germane to this discussion.

Crucial to the whole project is the development of a standardized terminology, for it determines how data items will be described both at the point of storing data in the system and at the point of accessing the information in the system. The DEIC Project Team feels that it is particularly important that any terminology used in the system reflects professional acceptance and is compatible with other educational information systems. This is not an easy goal for many reasons, one of which is the lack of agreement among professional sources. Nor is it a finite goal. Terminology is always in a state of change, and the need for a system that can accept continual updating has been a major consideration in the designing of this data system. We have decided to base our standardized terminology and working definitions on the U.S.O.E. handbook series, with reference to the ERIC Thesaurus, WICHE Management Information Systems Program technical reports, and Good's Dictionary of Education. For example, any financial statistics collected by SDE have been examined to see if terms defined in the federal handbook on finance and accounting (Handbook II) can be used to index the data items. The other sources mentioned are checked if there is confusion among terms or if there is no term in the Handbook to represent the concept. We work with blocks of subject-related data items, so that the standardized terminology will reflect subject relationships among terms within a subject area and so that clarification of scope or context can be provided as needed.

---

[2] Lind, A.D. "Automated Data Retrieval Interface Systems (ADRIS)." Interchange, March 1974, attachment #3, p. 2.

The product of this phase of the DEIC Project will be a thesaurus. One part of the thesaurus will display in outline form for various subject areas the scope, context, and interrelationships of all those terms that can be used to retrieve data on a particular subject. It will also indicate the codes to be used to retrieve the data desired. Part Two will be an alphabetical index of standardized search terms and synonymous terms that will direct the user to the appropriate place in the outline. The thesaurus will also include guidelines as to how the thesaurus is to be used (1) to enter data into the system, and (2) to search the system for specific information.

We already have segments of proposed terminology in first draft, and are entering a pre-test stage of the research, where we are applying our terminology and indexing system to a sample of forms distributed by SDE. We anticipate that reworking the vocabulary and debugging the system will be a lengthy and frustrating process. However, we are approaching it on a segment-by-segment basis; after one segment is tested and revised, the corresponding data can be entered into the system for immediate use. This process will also constitute the revising and updating procedure necessary to maintain a current and valid file of information.

What problems have we encountered? We've been plagued with all the usual challenges of creating standardized terminology -- synonym control, multiple meanings for single words, choosing from among several terms expressing the same concept, changing terminology. In addition, the data base itself has contributed its share of unique posers. Context is particularly troublesome. Removing the data item from the comfortable nest provided by the form often produces a baby bird with an identity crisis. Not only can it not "fly" alone, you can't even be sure whether you are observing a grounded sparrow or wren. For this reason we try to provide context by recording such supplemental information as collecting agency, coverage dates, and even additional subject information in the indexing terminology used to describe the data item.

Persons creating the forms tend to ask for information in phraseology they think the person filling out the form will understand. As a result, two conditions arise. The phraseology used on the form may express the need for information in circuitous rather than the straightforward terms used to record the data into the system. A very simple example of this is a frequent request for "The number of students in _____ program." With the possible exception of the name of the program, not a single word in that data item is significant for indexing purposes. Fortunately we can recognize quite easily the concept behind the request -- enrollment (it's not so simple in other instances). The fact remains, however, that the phrasing in the data item cannot be used for indexing purposes.

This is compounded when different forms may ask for similar information in different phraseology. For example, Form C may ask for total enrollment in the district's junior high school (incidentally, persons supplying this data from systems on 8-4, 6-2-4, and 6-3-3 plans may interpret this request differently), Form D may ask for the number of students in seventh and eighth grades within the system, while Form E may ask for projected enrollment in the first two years of high school in the district two years hence. All may be wanting essentially the same information.

For many good, and some not so good, reasons forms frequently become obsolete or are replaced with revised forms at SDE. A major concern of DEIC is file maintenance -- adding new data and reflecting that addition in the standardized terminology of the system, and deleting obsolete information and terminology from the system. For the revision component of the system, we intend to follow essentially the same procedure as is being used to develop the thesaurus. Rather than working with subject blocks, however, we plan to revise on a form-by-form basis; our file organization is flexible enough to provide for both subject and form approach. Our retrieval code is an open-ended one, so that additions to it do not require recoding of materials already in the system.

We see many contributions that DEIC can make to educational data control in New Jersey. We have already mentioned that standardizing terminology used in data collection and retrieval should result in more precision in filling out forms and therefore should provide more reliable data. In addition, the resulting thesaurus will be useful as a guide to filling out forms and accessing the data base efficiently and accurately.

Futhermore, the project will facilitate the creation of a computerized data base, accessible by any personnel authorized to examine the data. By coordinating the SDE data collection efforts within the framework of professional and national efforts, teachers at the local level and federal educational administrators alike can work with minimal additional instruction in the state or national educational information systems that are or may become available.

Finally, the vehicle used for data collection will be directly affected. In the future, phraseology used in requesting information should reflect the standardized terminology. As duplicate and synonymous requests are identified, there should be a corresponding reduction in the actual data items included on various forms. It is reasonable to expect that frequently "new" information needed by SDE would already be stored and thus retrievable from the system, eliminating the need for creating a new form to collect the information. A logical, though long-term, benefit of this project is computerized preparation of data-collection forms used by SDE. When direct computer access becomes available at the local level, there is the potential for directly requesting and entering data through the system as needed, thereby eliminating entirely the handling of forms.

A STATE LEVEL EDUCATION MANPOWER INFORMATION SYSTEM

by

John J. Stiglmeier, Ph.D.
Director
Information Center on Education
New York State Education Department

and

Michael E. Mirsky, Ph.D.
Assistant in Educational Data Systems
Information Center on Education
New York State Education Department

In 1972 a year-long study was undertaken by the New York State Education Department to plan for the development of a statewide information system based on existing education manpower statistical data. The study, supported by a U.S. Office of Education grant, was conducted by the Department's Information Center on Education. The results of this year-long effort have been compiled in a planning document entitled A Plan for the Development of a Comprehensive Educational Manpower Information System for New York State. The system (EMIS) is designed to encompass historical, contemporary and future data on educational manpower (EM) including public school classroom teachers, pupil personnel services specialists, and supervisory and administrative personnel. It contains data on individuals who have been certified in New York State since 1970 or who have been employed in New York State public schools during the same time span. EMIS is designed to provide precise and timely information in the following areas:

1) the existing educational manpower in public elementary and secondary schools;

2) the potential pool of educational manpower produced by teacher training institutions;

3) the potential pool of educational manpower included in the teacher certification files who are not presently employed in a school system;

4) the turnover rates of educational manpower in all sectors of the public elementary and secondary education system.

The system will serve as a base for the planning and development of professional education manpower training programs in the colleges and universities of New York State. These institutions produce three-quarters of the State's professional educational personnel.

Rationale for the EMIS

As recently as 1966 the Nation was faced with a shortage of certified teachers. The Senate Committee on Labor and Public Welfare and the House Committee on Education and Labor issued reports stating that in 1966 the Nation was faced with an unprecedented shortage of almost 170,000 qualified teachers. (Zerfoss & Shapiro, 1973). Since that time, however, record numbers of teachers have graduated from colleges and universities and many school districts now report a substantial surplus of applicants for teaching positions. The National Education Association annual report on the supply and demand for teachers clearly shows that since 1970 the supply of teachers has exceeded the demand. Several reasons appear to explain this rather sudden change in the supply and demand situation. They include: (1) a sharp increase in the number of college graduates at the end of the 1960's; (2) a slowdown in the growth of the school age population; and (3) a tightening of fiscal constraints in many school districts. The New York State picture of the teacher oversupply situation is reflective of the situation at the National level.

It has become increasingly evident that manpower planning in the area of educational personnel production and utilization in New York State must be brought under systematic control. The key to solving this problem is the development of a statistically-based information system which can routinely generate data reflecting such central factors as (1) the rate of production of EM by institution and area of certification and (2) the rate of employment and utilization of EM. The EMIS will serve as a readily available base of information whose components have been identified by potential users as necessary for sound educational manpower planning.

Development of the EMIS

Two pragmatic principles were adopted in guiding the development phase of the EMIS. They were (1) potential users of the information should help define policy issues and, in turn, the type of information to be generated by the system and (2) existing data sources should be used as the base for system development. In line with these principles, the first developmental task was the identification of key policy issues relating to efficient and systematic production and utilization of professional educational manpower. The identification of policy issues then led to the identification of required statistical information which would bear directly on the resolution of these issues and serve user needs. Next, relevant data sources contained in presently operating systems, both within and outside the New York State Education Department, were identified. A match was then made between those systems and the data requirements identified earlier to determine what, if any, new sources of information (or modifications of existing sources) would be required to adequately address pertinent EM policy issues.

The policy issue identification phase of the study was undertaken by conducting in-depth interviews with individuals in various sectors of the State's educational system, including the State Education Department, private and public colleges and universities, local school districts, education planning groups and National and State teacher affiliated groups. The major thrust in identifying education manpower policy issues was to find areas in which there are gaps in existing information and where lack of communication exists among various sectors of the education establishment, thereby hampering efficient EM planning. A content analysis of the interview data surfaced a set of significant issues and information needs. The most critical issues were those related to the need for an information base that would interrelate data on characteristics of manpower produced by the colleges and universities and the employment of such manpower in the public elementary and secondary schools of New York State. It was determined that such a comprehensive picture would increase the likelihood of communication between groups and serve as a basis for more efficient EM program planning.

The results of the first phase of the study also indicated that unit record data on individuals would be required as the primary data source in the system. A survey was conducted to identify the current or potential availability of such information. Three sources of unit record information were identified which could serve as a data base for the EMIS. They are: (1) the Basic Educational Data System (BEDS); (2) Teacher Certification Records (TCR); and (3) New York State Teacher Retirement System Files (TRS).

Basic Educational Data System (BEDS)--BEDS, which has been operational since 1967, is a computerized system operated by the New York State Education Department. The personnel file of the system identifies, on a longitudinal basis, all public school professional personnel. Each unit record contains a series of demographic and professional characteristics and the assignments that each staff member performs. BEDS, then, serves as the base for identifying the past and existing public school professional EM pool employed in the State.

Teacher Certification Records (TCR)--The Education Department maintains a manually operated filing system on all individuals to whom professional education certificates have been issued. Demographic and professional characteristics on each individual are compatible with data contained in the BEDS personnel file. In order to incorporate this data source into the EMIS it was necessary to convert the manual system to a computer file and to create and install a system by which new records would be made a part of the system on an ongoing basis. The initial data conversion effort involved approximately 6.5 man years of clerical staff handling in excess of 322,000 records, accounting for all certificates issued in New York State since 1970. The second procedure involved implementing a new administrative procedure in the Department's Division of Teacher Certification.

Computer programs have been written which merge the BEDS files with the TCR files. This newly combined data base, which contains information dating back to 1970, can produce information which shows the relationships between program output and school district utilization at the district, county, region and State levels. Information can be sorted on any single data element contained in the system or any combination thereof.

New York State Teacher Retirement System Files (TRS)--This computerized system, which is independent of the Education Department, has a membership composed of the majority of professional personnel who are presently employed in or who have retired from service in the public elementary and secondary schools in New York State, excluding New York City. It has a total membership of 216,500 individuals. New York City operates its own teacher retirement system.

The master file of the TRS is partitioned into an active and retired file. Individuals upon retirement or death are removed from the active file when they leave active service. If the individual returns to active school service on any basis, he is again placed on the active file. Status changes for personnel such as retirement, death or reentry into service are periodically updated on the computer file. All retirees are identifiable by name, social security number, date of retirement, and school district from which they retired. The same type of information can be generated from the New York City Teachers Retirement System.

The data elements contained in this system are compatible with BEDS and TCR files. The TRS can provide unit record data on the rate at which professional personnel are leaving the field as active members of the public education field thereby providing a more accurate picture of the potential pool of certified EM who are not employed. Inclusion of the TRS files in the EMIS is yet to be arranged. This will require negotiations between the Education Department and the Teacher Retirement System.

Utility of the EMIS

Output from the EMIS can serve many sectors of the New York State Education community in terms of more efficient EM planning. At the State level, trends in the supply and utilization of specific types of EM on a statewide and regional basis will highlight present and anticipated areas of EM growth, contraction and stability. At the local school district level, superintendents will be provided with timely information for locating appropriate EM to fill vacancies or newly created positions. They will also have information available on general trends in EM availability and needs in their region and in the State.

Guidance counselors in both secondary schools and colleges will be able to provide students with relevant and up to date career information dealing with trends in the growth, stability and decline of EM employment. Colleges and universities will be able to plan more accurately for the creation, expansion, contraction or elimination of programs dealing with the production of EM.

Another important application of EMIS will be to provide a statistical base for generating more reliable forecasts of future demand for all categories of certified EM. The projections can be computed on the basis of such factors as certification field, geographic region, type of training institution, type of district or any other data element within the system. The system is designed to be flexible in terms of output format. A combination of general statistical outputs on a statewide and regional basis will meet most planning information needs. When necessary, special reports will be generated on a request basis.

The EMIS is designed to provide relevant education manpower planning data to more adequately address problems of supply and demand in the present and in the future.

A data flow outline for the complete system is shown in Figure 1. A listing of the data elements contained in the system is shown in Figure 2.

Figure 1

DATA FLOW OUTLINE

NEW YORK STATE EDUCATIONAL MANPOWER INFORMATION SYSTEM

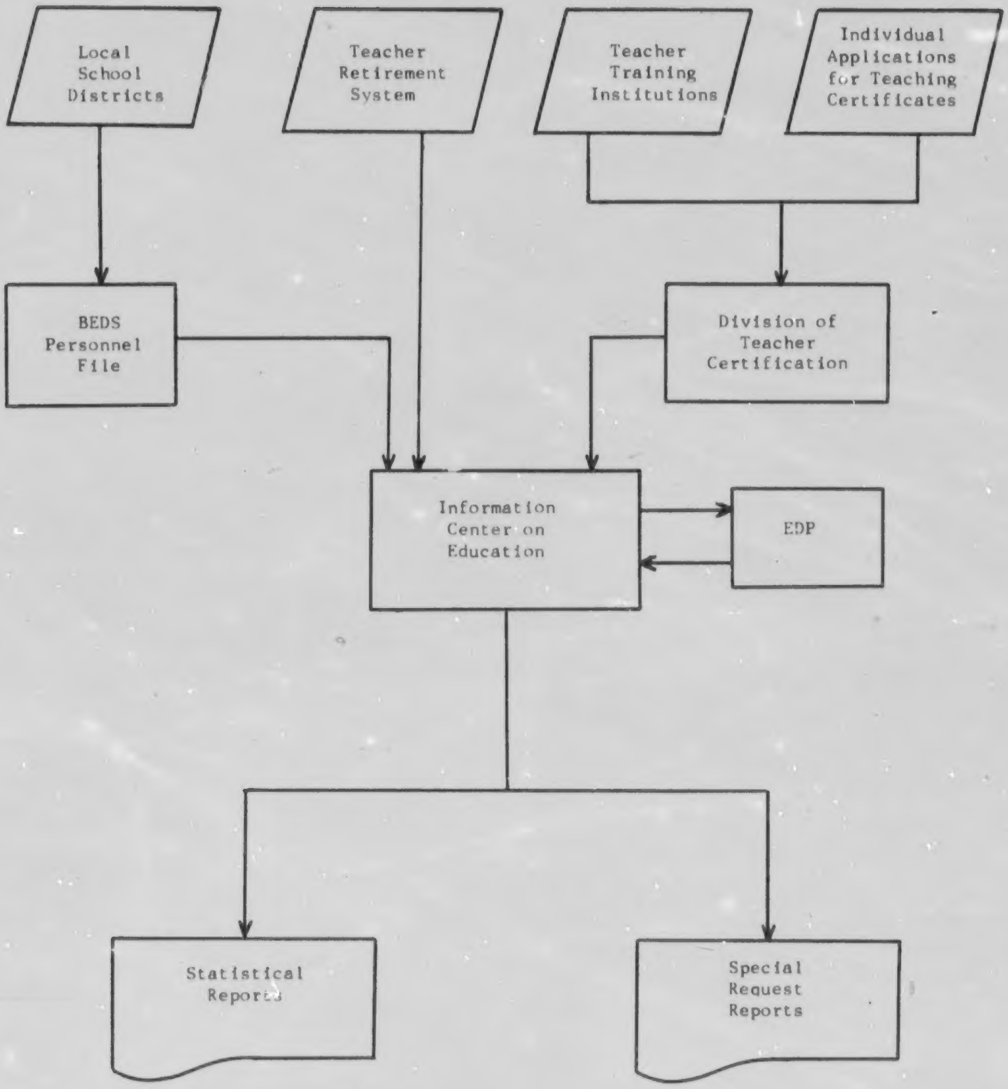

Local School Districts
Teacher Retirement System
Teacher Training Institutions
Individual Applications for Teaching Certificates
BEDS Personnel File
Division of Teacher Certification
Information Center on Education
EDP
Statistical Reports
Special Request Reports

Figure 2

MAJOR DATA ELEMENTS CONTAINED IN THE EMIS

Bureau of Educational Data Systems	Teacher Certification Records	Teacher Retirement System
Name	Name	Name
Social Security Number	Social Security Number	Social Security Number
Date of Birth	Date of Birth	Date of Birth
Sex	Sex	Sex
Marital Status	For Each Certificate	Work Status
Degree Status	College (Name and County of Location)	Active
Occupation in Prior Year (nature and location)	Certification Area	Retired
Tenure Status	Year of Certification	School District
Educational Experience	Type of Certificate	
Certification Status		
Professional Field Assignments		
School District of Employment		

INVERTED FILE STRUCTURES
IN PAYROLL PROCESSING

Thomas C. Richards
Miami University Oxford, Ohio

Introduction

A great deal of attention has recently been devoted to the subject of generalized data base management systems (4,6). This interest has occured in parallel with the emergence of computer based management information systems (2). These systems are the backbone of reporting systems in government and industry which provide information for management decisions with the necessary flexibility, adequacy, and frequency.

Proper file structure, selection and evaluation are of the utmost importance in view of this ever-growing data retrieval and reporting demand. A data structure must permit a direct representation of the complex relationships which exist among the entities of a business enterprise and permit many different logical orderings to be defined instead of the single ordering implied in a hierarchical structure.

The purpose of this paper is to discuss the successful design of a payroll-peronnel system which provides many of the above attributes while at the same time being implemented on a small core computer with a limited number of peripheral storage devices. This data base management system uses a unique file structure which allows extreme flexibility in file manipulation and retrieval. This unique system has actually been implemented for a medium size school district.

The Problem

There is a real and recurring need in educational administration to be able to interrogate a data base by selecting or extracting some part or combination of elements. For example, how many fourth grade teachers have masters degrees and are partially supported by Title III funds, or what is the average salary of a tenured teacher? The unit costs of schools, programs, and functional areas are all of importance to decision making processes in educational administration. Since the predominant costs of a public school district are personnel salaries, the area of payroll-personnel systems offers the greatest payoff in terms of providing meaningful cost analysis and cost benefit data.

Personnel file structures in public education tend to be rather straight forward since the teacher population of a school district is constant during the school year. Some categories of the classified staff tend to move in and out of employment but this is a rather minor turnover. The problem arises in the payroll file structure since the funding of a public school district tends to be rather complex.

The average public school district's funding depends on a multiplicity of sources of funds including local taxes, state subsidies, and a variety of federal programs. Personnel within a school district may be paid from several sources and detailed accounts must be kept of the allocation of funds from each source. Thus, if an employee's salary is drawn from more than one fund, a detailed payroll record must be kept for each fund. The problem is further complicated when his salary is charged against several accounts within a fund.

The Solution

One obvious solution would be to construct a payroll-personnel record for each employee so that current and year-to-date payroll information can be carried in each record for each possible fund and account. Unfortunately, the employee whose salary must be allocated against more than one fund is an exception rather than a rule. This method of attack would lead to an excessive record length for most employees. Another solution used by many school districts is to generate a separate payroll record for each fund. This results in a great deal of duplicate information since each record usually contains employee name, address, social security number, number of dependents, and other items common to all records for a particular employee.

The solution used in this investigation is to carry a master record for each employee which contains all of these common elements plus the multiplicity of information required for employee personnel records on a random access file. By using the coding structure outlined below, this master record is then linked into one or more payroll records. One of these records is generated for each fund from which the employee's salary is drawn. All employees have a master record and one payroll record in the file as shown in Figure 1. The exception has a master personnel record and two or more payroll records.

One additional problem can be handled with ease using this file structure. This is the case where a salary is charged against two or more accounts within one fund. If each payroll record has an account number associated with it, each account within the same fund generates a separate payroll record for the employee. For example if an employee's salary is drawn from general funds and ESEA funds, two payroll records are generated. An employee whose salary is drawn from DPPF funds and charged against librarian salaries and bus driver salaries would also have two payroll records, one for each account within the DPPF funds.

This inverted file structure allows a great deal of flexibility in file content. For example, it is often necessary to carry a particular type of information for one classification of employee which will never be required for another employee type. For certificated school employees, it is often necessary to maintain, in machine processable form, certification data, classroom assignment data, and the number of semester hours completed. This information would not be required for classified employees. If room were left in the master record for all of this information, the file length would be exorbitant in length. Likewise, it might be necessary to carry a job skills inventory, training information or civil service examination data for classified personnel. By using the same coding structure used to link the payroll records into the employee master record, it is possible to carry a multiplicity of specialized supplemental records containing a large variety of information.

The coding structure required for this file organization is suitable for random access processing. Each employee's master record contains, as a primary key, his identification number. The detail payroll record uses as its key a combination of this identification number, a fund number, and an indicator telling how many accounts within the fund his salary is debited or charged against as shown in Figure 2. The employee master record contains, in addition to his identification number, up to six fund identifiers in a sub-record. Each field within this sub-record contains, as shown in Figure 2, a fund number plus an account indicator.

The fund and account indicator are then combined with the employee identification number to generate the key for each of his payroll records. By using a dummy fund number, it is also possible to store and retrieve miscellaneous information in a dummy payroll record. Using an index sequential file and keeping all records the same length, it is possible to access the file several different ways. The employee identification number can be used as an accession key to retrieve all payroll information related to a particular employee. The fund number can be used as the sequential accession key to retrieve all records related to a particular fund. By retrieving all payroll records and then sorting by account number, it is possible to easily generate account distribution reports.

Conclusion

The personnel-payroll data base management system outlined in this report offers the flexibility and adequacy required for implementation of useful and powerful systems on a small core computer with a limited number of peripheral storage devices. In particular, this system has been installed on a 32K IBM 360 model 30 with two disk drives and two tape drives using COBOL as the source language. The file structure and additional data elements and records can be added as user requirements change. The system allows direct representation of the complex relationships which must be analyzed for effective administration of a payroll-personnel system in educational administration.

References

1. Berztiss, A. T., Data Structures - Theory and Practice, Academic Press, New York, 1971.
2. Cagan, Carl, Data Management Systems, Melville Publishing Co., Los Angeles, 1973.
3. Cardenas, Alfonso F., Evaluation and Selection of File Organization - A Model and System. Communications of the ACM, Sept, 1973, Vol 6, No 9.
4. Codasyl Systems Committee, Introduction To Feature Analysis of Generalized Data Base Management Systems. Communications of the ACM, May, 1971, Vol 14, No 5.
5. Dodd, George G., Elements of Data Management Systems, Computing Surveys, June, 1969, Vol 1, No 2.
6. Schubert, Richard F., Directions in Data Base Management Technology, Datamation, Sept, 1974, Vol 20.
7. Severance, Dennis, G., Identifier Search Mechanisms; A Survey and Generalized Model, Computing Surveys, Sept, 1974, Vol 6, No 3.

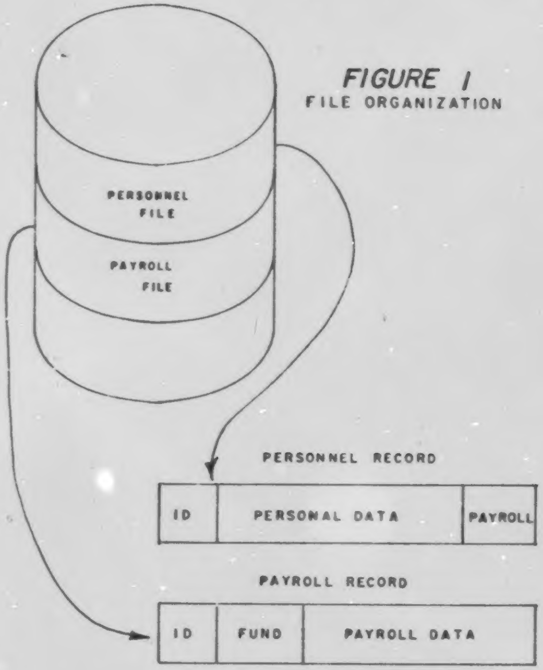

PERSONNEL FILE
PAYROLL FILE
PERSONNEL RECORD
ID
PERSONAL DATA
PAYROLL
PAYROLL RECORD
ID
FUND
PAYROLL DATA

FIGURE I
FILE ORGANIZATION

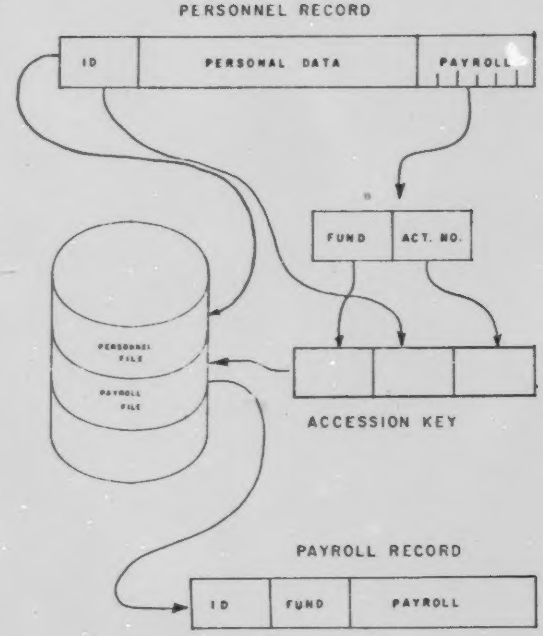

PERSONNEL RECORD
ID
PERSONAL DATA
PAYROLL
FUND
ACT. NO.
PERSONNEL FILE
PAYROLL FILE
ACCESSION KEY
PAYROLL RECORD
ID
FUND
PAYROLL

FIGURE II
PAYROLL RECORD ACCESSION

THE NEW WORLD OF INTERACTIVE BUSINESS SYSTEMS

Edward J. Lias
Ocean County Information Network

The conference theme "New Worlds of Educational Data Systems" may be an exciting headline to the novice or the outsider looking in. But the serious business administrator who has invested years of design in his current office systems may not be cheered to hear of bold new approaches to educational data systems. New worlds can be seriously threatening.

The theme before us this hour may therefore be of major significance. Interactive business systems, now operational, could impact your life or office systems during the next five years.

The contemplation of such change is disturbing--maybe unwelcome. It represents a new style of processing and could unsettle established conventions in the future.

Whether you read "Business Week" or "Computer World," you are probably aware that terminals are here to stay. Four years ago they cost $5,000 each. Some now cost $1,200. They weighed (typically) 70 pounds in 1970. Today this one (DIGILOG) weighs 9 pounds and goes easily in a brief case. You can take it home and attach one wire to the rabbit ears of your TV set which serves as a screen.

About two million terminals (all types) are installed around the world. The companies which make them are tooled to supply them like telephones for the next 50 years. Each office and home is a potential buyer. In 1973-74, two new terminals were introduced each week.

Till now, terminals had little to do with your office automation. The mature needs of a school administrator had nothing in common with the superficial services which terminals typically provide.

Here are some common uses of terminals:

1.	ONLINE UPDATING	AIRLINE RESERVATIONS.
2.	DATA BROADCASTING	AIRLINE FLIGHT SCHEDULES
3.	MESSAGE SWITCHING	WESTERN UNION TELEGRAM SERVICES.
4.	DATA ENTRY	ONLINE KEYPUNCH REPLACEMENT. KEYPUNCHING ONTO CASSETTE, TAPE, OR DISK.
5.	INQUIRY - RESPONSE	EXECUTIVE VIEWING OF FINANCIAL DATA.
6.	R.J.E.	EXTENDS COMPUTER ROOM TO ANOTHER PLACE. PROCEDURES SAME AS IN COMPUTER ROOM.

None of the terminal uses above are complete business systems sufficient to drive the full data requirements of an active school system. An interactive business system will be a combination of several of the above, plus much more.

As an illustration we will detail a student record-keeping system. Please observe that the illustration could be payroll, budget, inventory, attendance, or any other business system. But by using the student record-keeping system, you can relate your current system to the one shown here.

OASIS

COMPUTER TASKS AVAILABLE
(PUBLIC SCHOOLS)

02 Add New Students to Master file
04 Add/Drop View Student Courses
05 Add/Drop courses by class
06 Add - Bus number, homeroom, district numbers, etc.
07 Sort Student Courses into Course/Period order

CHANGE PROGRAMS (10-19)

10 Change Student Facts
17 Calculate Student's Age

VIEW PROGRAMS (20-29)

20 View Student Information
27 View Course Cross-Reference Keys
28 View Cross-Reference Keys

REPORT CARD PACKAGE (30-40)

30 Batch Grade Entry
31 Grade Entry to Student Records
32 Check for Missing Grades in Master File
33 Calculate Grades + Attendance for Report Cards
34 Print Report Cards
35 Print Incompletes + Failures by Teachers
36 Print Incompletes + Failures by Students
37 Print High + Regular Honor Rolls
38 Print Grade Distribution
39 Print Class Grades Entered
40 Write Grades to Student Records

INITIALIZATION PROGRAMS (41-50)

41 Build Master Course File
42 Modify Master Course File
43 New Year Student Requests for Courses
44 Build/Print Teacher Schedules
46 Create Next Year's Master Student File
50 Print Student Schedule Cards (Beginning School Year)

REPORT PROGRAMS (51-69)

51 Total Students in Each Grade
52 District Statistics
53 Print Commendation/Deficiency List by Teacher
54 Individual Entries of Commendation/Deficiency Students
55 Print Commendation/Deficiency Report
56 Print Total Com/Def By Grade (Audit)
57 List Deficiencies for Counselors
58 Formatted Print of Students
59 List Failures Without Deficiencies
60 Com/Def Statistics Report
61 Bus/Pupil Mileage
62 Student ID Listing
64 Print Class List by Teachers
66 Print End-of-Year Labels
67 Total + Print Student by Courses

ATTENDANCE ENTRY AND UPDATE PROGRAMS (A1-A9)

A1 Entry of Daily Attendance
A2 Print Daily Attendance List
A3 View Individual Attendance
A4 Attendance Officers List
A6 Change Attendance Data
A7 Change Enrollment Code
A8 Change Status Data
A9 Correct Single Status & Attendance

ATTENDANCE IN-HOUSE PROGRAMS (B1-B9)

B1 Print Attendance Labels
B2 Print Attendance Cards for Teachers
B3 Build Control Record Calendar (School Calendar)
B4 Modify School Calendar
B5 Print School Calendar

ATTENDANCE REGISTER AND ANNUAL REPORT PROGRAMS (C1-C9)

C1 Monthly Attendance Register
C2 Annual Attendance Register
C3 Age Distribution of Transfers Coming In For Annual Report

SCHOOL HISTORY PROGRAMS

H1 Initialize History File for First Time
H2 End of School Year Update
H3 Display Individual or Class History File Record
H4 Add, Drop, Modify or View Courses on the History File
H5 Delete Records from the History File
H6 Clean up History File - Elim Drops + Transfers
H7 Calc Final Grade for First Half School Year
H8 Calc + Print Any Grades Class Rank from History File
H9 Add New Students + Change Students' Facts on History File
T1 Sort Courses on History File
T2 Accumulative Class Rank
T3 Check History File for Missing Courses

CONTROL RECORD MAINTENANCE PROGRAMS (80-89)

86 Modify Cross-Reference Keys
87 Build Cross-Reference Keys
88 Table of Task Contents

TEST AND REORG PROGRAMS (90-99)

90 Adjust Any Data Element
94 Create Test File Indiv. Stu. or Courses
95 Create Test File
98 Delete Records
99 Reorg Student Master File

EXHIBIT 2

Assume that a terminal is in the office next to you and that one clerk or secretary has been trained to type the few lines required.

From the list of tasks we will choose number 2. The tasks are all the functions which can be performed by the package. (See Exhibit 2.)

When task 2 is requested, a small program loads to perform the function of receiving new students who are just entering school. Each task is 4 to 8 thousand characters in length (very small).

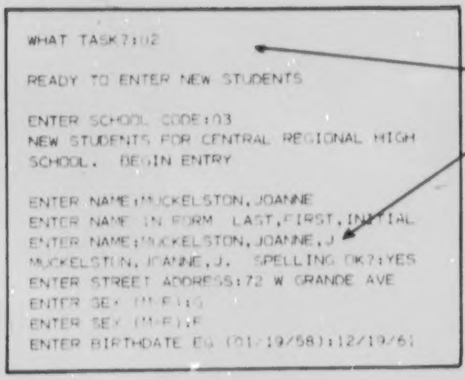
Task 2 says:

At first the terminal is a data entry device. But when the NAME is entered it becomes a file searcher, rejecting your entry if the name is already on the file. On significant data it echos back the data for verification--but does not require double typing for verification; only the letter y for "yes" is typed.

What is an Interactive Business System? A system in which <u>every</u> <u>needed</u> <u>procedure</u> <u>of</u> <u>an</u> <u>application</u> <u>is</u> <u>available</u> <u>on</u> <u>a</u> <u>terminal</u> in the local office.

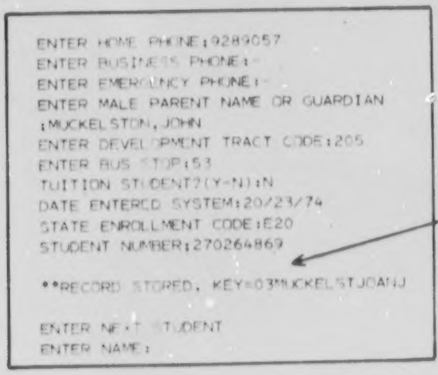
If you select task 20 or 30 or 64, you will be eliciting your own scholastic reports. The report programs may ask, "Do you want this report on the terminal or the high-speed printer? Do you desire single or double spacing? How many copies do you desire? Sorted by bus number or homeroom?"

Upon receiving these controlling facts, the file is scanned by the program, and the report is generated automatically. No one at the Computer Center is called or involved in the run.



```
WHAT TASK?:02

READY TO ENTER NEW STUDENTS

ENTER SCHOOL CODE:03
NEW STUDENTS FOR CENTRAL REGIONAL HIGH
SCHOOL.  BEGIN ENTRY

ENTER NAME:MUCKELSTON,JOANNE
ENTER NAME IN FORM  LAST,FIRST,INITIAL
ENTER NAME:MUCKELSTON,JOANNE,J
MUCKELSTON,JOANNE,J.  SPELLING OK?:YES
ENTER STREET ADDRESS:72 W GRANDE AVE
ENTER SEX (M-F):G
ENTER SEX (M-F):F
ENTER BIRTHDATE EG (01/19/58):12/19/61
```

Task 02 illustrates user-controlled file management. Here a <u>new</u> <u>student</u> is added to the file by trained clerks who conversationally guide the process. The program is 4K in length and error checks all possible field criteria.



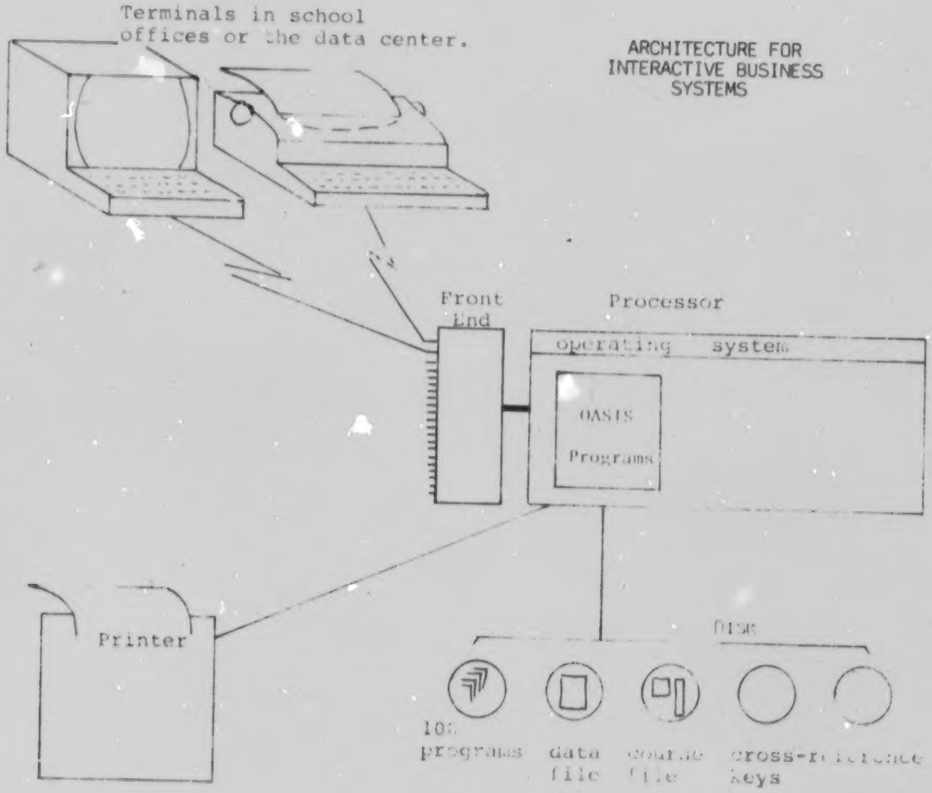
```
ENTER HOME PHONE:9289057
ENTER BUSINESS PHONE:-
ENTER EMERGENCY PHONE:-
ENTER MALE PARENT NAME OR GUARDIAN
:MUCKELSTON,JOHN
ENTER DEVELOPMENT TRACT CODE:205
ENTER BUS STOP:53
TUITION STUDENT?(Y-N):N
DATE ENTERED SYSTEM:20/23/74
STATE ENROLLMENT CODE:E20
STUDENT NUMBER:270264869

**RECORD STORED. KEY=03MUCKELSTJOANJ

ENTER NEXT STUDENT
ENTER NAME:
```

If the clerk types STORE, the record could be stored without these less critical elements.

The record is stored <u>and</u> 3 cross-reference keys are stored including student #, bus #, development #.

Exhibit 3 diagrams an Interactive Business System. It is a new world of simplicity. Gone are sorts, merges, batch updates, transaction-proof lists, etc.

ARCHITECTURE FOR INTERACTIVE BUSINESS SYSTEMS


Terminals in school offices or the data center.
Front End
Processor
operating system
OASIS Programs
Printer
Disk
105 programs
data file
course file
cross-reference keys

(SMALL REPORTS CAN BE RECEIVED ON THE USER'S TERMINAL WHEN DESIRED. OTHER REPORTS ARE DIRECTED TO THE HIGH-SPEED PRINTER. NO CARD PUNCH IS REQUIRED.)

Please lay this diagram, mentally, over your current applications. It can fit any of them. The terminal can ask payroll questions (What task? Add new employees, change wages, print W-2 forms, etc.) or school questions (What task? Drop courses, change attendance, print report cards, etc.)

Note that when the record is sent to disk for storage, it is already placed in key sequence (master alpha sorting is done) and that all necessary cross-reference keys are built at that time. One second later all varieties of reports can be generated. Thus, no merging or sorting is ever required.

What hardware permits this to be done? Most medium-size machines have common facilities which encourage this type of system, but not many application designers have utilized them. Diagrammed above is the UNIVAC 70/46 which hosted the first development of these systems.

Its remarkable operating system made the software development easy. VMOS (Virtual Memory Operating System) encourages online implementations. Note that it has no card punch. As a medium-size machine, it commonly supports 30 terminals, all of which may be running various Interactive Business Systems.

Reaching to the key question, my address in a sentence asks, "Who benefits in the new world of Interactive Business Systems?"

Several beneficiaries can be identified.

I. The Local Business Office

1. Your office might benefit from the disarm-simplicity.

As businessmen, you have probably looked askance upon DP operations in general, saying secretly, "Is it really necessary for people to be dashing in and out of the machine room--cards a-flying? Why so complex? Are all the awaited merges, proof lists, cylinder allocations, etc. really required?

In truth, batch systems are difficult to simplify. You can't do it with fewer than 12 separate steps, each of which awaits the completion of the one before.

But in the new world described here, your local office staff will run the application after two or three days training in the task numbers and what they do.

2. Your local office might benefit from the superior control as well.

Rather than dramatize those occasions when you felt that data processing was controlling your office, when you wished you were in charge of the data events--rather than surmise, let me describe a few (of many) instances where local office control is evident.

The College Registrar

The College Registrar has terminals in his office. Frequently he needs to know seat status or enrollment facts, or student lists. Without central operations being involved, he requests his reports when he wants them through his terminal. Since they are often bulky, he generally directs them to the high-speed printer and picks them up during lunch hour. All dropping or adding of courses is done on his terminal. Since the file is always up to date and always sorted, any desired report will reflect current status as of one second ago.

Attendance

A high school clerk types in the names of the absentees. Then, on a typewriter terminal they take task A2. It tells them to insert a ditto stencil, after which it prints the absentee list alphabetically and separated by grades. They do it morning and afternoon. The number of consectuive absences is listed for each person. The list is in all teacher mail boxes by 9:30 or 10:00 each day. The Computer Center is not involved in the operation at all. The monthly attendance registers result from task C1 with equal ease.

Discover the new world of simplicity and local control, available in your local office.

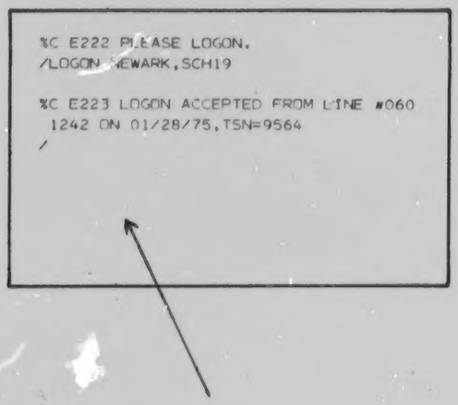
II. The Computer Operations Area Also Benefits

If a school superintendent visits the Center and says, "I think I'd like to try your system," after some conversation regarding the requirements, the following procedure would set it up in about three minutes:

1. LOGON. 2. Execute the SIPO package. 3. Enter the control facts. 4. Begin to enter students.

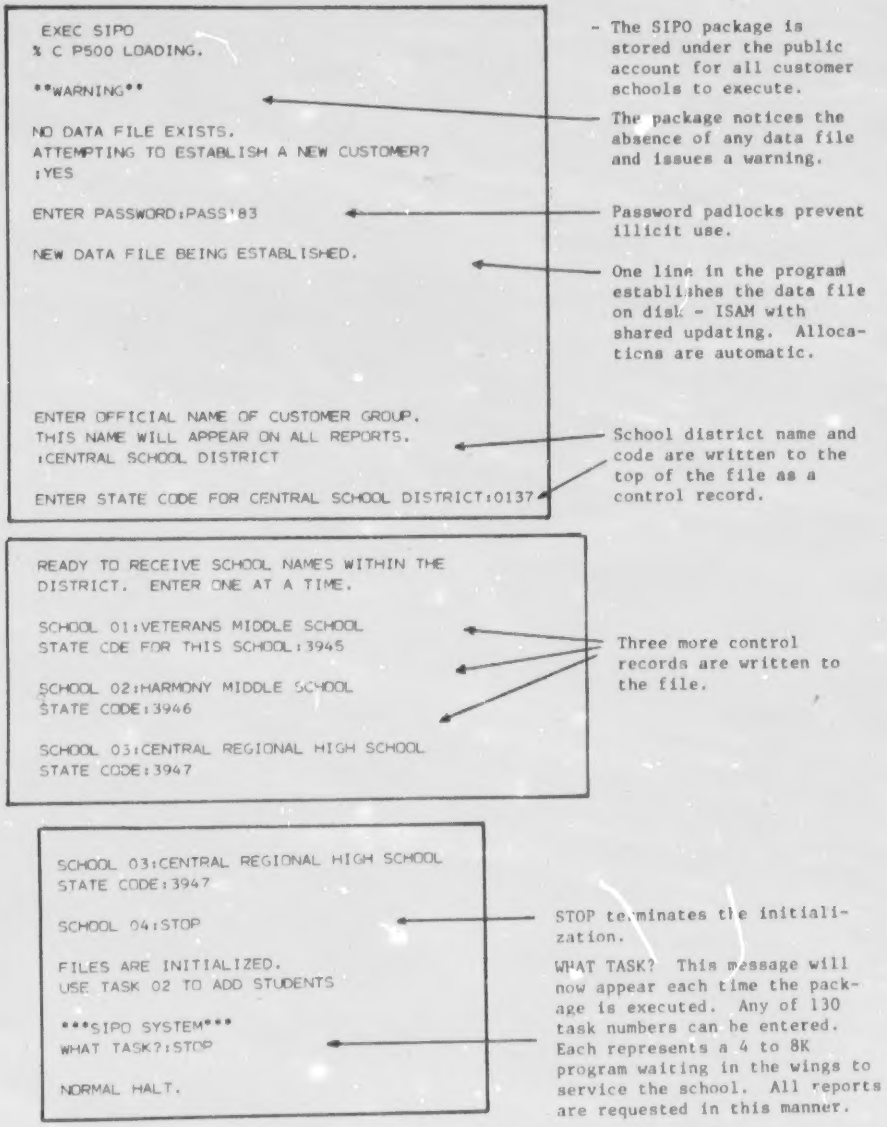
If at your site it takes two or three days to set up a new customer, this is justifiable--You may have track and cylinder allocations to calculate, various file formatting strategies to anticipate, sorting procedures to define, etc.

However, the procedure below is so simple, so free of operator intervention, and so automatic, that you could do it yourself.



```
%C E222 PLEASE LOGON.
/LOGON NEWARK,SCH19

%C E223 LOGON ACCEPTED FROM LINE #060
 1242 ON 01/28/75,TSN=9564
/
```

A visiting school superintendent thinks he would like to carry his school records on your machine. Ask your system administrator to join an account using the name of his city. While he is present, LOGON to the account immediately. Note that no programs or data files exist under this account.



```
 EXEC SIPO
% C P500 LOADING.

**WARNING**

NO DATA FILE EXISTS.
ATTEMPTING TO ESTABLISH A NEW CUSTOMER?
:YES

ENTER PASSWORD:PASS'83

NEW DATA FILE BEING ESTABLISHED.




ENTER OFFICIAL NAME OF CUSTOMER GROUP.
THIS NAME WILL APPEAR ON ALL REPORTS.
:CENTRAL SCHOOL DISTRICT

ENTER STATE CODE FOR CENTRAL SCHOOL DISTRICT:0137
```

- The SIPO package is stored under the public account for all customer schools to execute.

The package notices the absence of any data file and issues a warning.

Password padlocks prevent illicit use.

One line in the program establishes the data file on disk - ISAM with shared updating. Allocations are automatic.

School district name and code are written to the top of the file as a control record.

```
READY TO RECEIVE SCHOOL NAMES WITHIN THE
DISTRICT.  ENTER ONE AT A TIME.

SCHOOL 01:VETERANS MIDDLE SCHOOL
STATE CDE FOR THIS SCHOOL:3945

SCHOOL 02:HARMONY MIDDLE SCHOOL
STATE CODE:3946

SCHOOL 03:CENTRAL REGIONAL HIGH SCHOOL
STATE CODE:3947
```

Three more control records are written to the file.

```
SCHOOL 03:CENTRAL REGIONAL HIGH SCHOOL
STATE CODE:3947

SCHOOL 04:STOP

FILES ARE INITIALIZED.
USE TASK 02 TO ADD STUDENTS

***SIPO SYSTEM***
WHAT TASK?:STOP

NORMAL HALT.
```

STOP terminates the initialization.

WHAT TASK? This message will now appear each time the package is executed. Any of 130 task numbers can be entered. Each represents a 4 to 8K program waiting in the wings to service the school. All reports are requested in this manner.

With Interactive Systems there is no JCL. No control card decks. No program patching or recompiling. Freed from heavy operations effort, the center can concentrate on customer training, etc.

The benefit being described here is reduced activity in the machine room. If the computer can set up the system, why not let it do it? No one in operations needs to be called. Audit trails on the console show that it was done, but no one stood by to make it happen.

Since the file is maintained in the major sort sequence and the other sorts are maintained in the cross-reference keys, any report can be printed at any time.

With the superintendent still there, you can begin to enter a few students just to show him. If you recall, we reviewed as task 2.

In the world of Interactive Business Systems, those who automate other people's offices will automate their own shop as well. Modern operating systems encourage it and Interactive Business Systems make it happen.

III. The Costs are Better Than You Might Think

There is a general impression that the flashy terminals, phone lines and central machine are far more costly than a local batch shop. Most such beliefs are based on throwbacks to 1970 analyses. Four years later, such paleolithic findings are misleading. This low cost terminal did not exist four years ago! Four years ago the central hardware was twice as costly to purchase as it is today. (Get the price on a UNIVAC 90/60V or an IBM 370/145.)

Not everyone knows about economy of scale. Three dozen eggs are cheaper than one, we know. But where in life is "cheaper by the dozen" most magnified? Not in selling three Brooklyn Bridges; not in the World Trade Center, but in the computer market. It is not a trick of pricing, but a fact of electronics. You can move ten characters around for a penny in a small computer. You can move 10,000 characters for a penny in a larger one. In this one you can move about 10,000 characters for a penny. Next size bigger, 100,000 characters for a penny. The really big and fast machines will reach a million characters/second.

ECONOMY OF SCALE

Small --------	1	100 Char/Penny
	2	1,000 Char/Penny
	3	10,000 Char/Penny
Medium --------	4	100,000 Char/Penny
	5	1,000,000 Char/Penny
Large -------->		
	6	10,000,000 Char/Penny

If you come from a small batch shop, you may be in the ten or 100 characters/penny class.

Similarly, the economy of personnel is surprising. Since the activities in the machine room are automated, the computer center staff will be more oriented to system development than to the brush fires of operations. Here is a comparison.

A few people will debate that Exhibit 7 omits some of the steps in the batch process. None are omitted on the interactive side, however. On batch side some of the steps take five minutes; others take half an hour.

The batch procedures are mostly a manual sequence of events within the computer shop. Eleven human events compared to two translates into improved use of human skills on the interactive side. People are freed to perform customer relations or improved system design. The personnel costs can be significantly lower per throughput volume on the interactive side because the machine is doing more of the work. They who automate other people's offices should automate their shops internally as well.

FUNCTIONS PERFORMED

BATCH SYSTEMS	INTERACTIVE SYSTEMS
1. TRANSPORT FORMS TO COMPUTER CENTER	1. COMMUNICATE DATA TO COMPUTER CENTER
2. ASSIGN DATA TO KEYPUNCH	
3. KEYPUNCH THE DATA	2. TYPE THE DATA ON THE TERMINAL
4. VERIFY THE DATA	
5. LIST THE PUNCHED CARDS	
6. WRITE THE PROGRAM TO PROOF THE CARDS	3. WRITE THE PROGRAM TO RECEIVE DATA FROM TERMINAL
7. SCHEDULE AND RUN ABOVE PGM	
8. STUDY OUTPUT FROM ABOVE RUN	
9. CORRECTION CYCLE (S)	
10. WRITE PGM TO MERGE CARDS TO MASTER FILE	
11. RUN THE FILE UPDATE PGM	
12. MAYBE PUNCH MASTER FILE TO CARDS	4. CREATE SAVE TAPES OF DISK FILES
HUMAN TASKS: 9	HUMAN TASKS: 2
ELECTRONIC TASKS: 3	ELECTRONIC TASKS: 2

EXHIBIT 7

In this address I have attempted to show the benefits of Interactive Business Systems as experienced in the local offices, the computer center, and in the cost effectiveness.

Is it a fad? Have systems like these any permanence in the evolution of business systems? Ponder this question as you shortly move to other sessions to witness other similar systems.

My personal opinion is that man as an information processing creature is moving in a known direction. Inherent in the invention of the alphabet was the eventual discovery of the printing press. And the printing press was certain to create information overload. With more words and paper tumbling upon us than we can read or file, electronics was destined to relieve this very human need.

But online audio systems have been accepted by every teenager and housewife, the telephone being a global network for exchanging audio data.

Without conscious theorizing, most people can make the obvious assumption: "My character data should be as easily manipulatable on some telephone-like device as my vocal data." (See Exhibit 8.)

Terminals and Interactive Business Systems represent a firm step down the path of our information instincts. Exhibit 8 details these instincts and summarizes at the bottom my belief in the increasingly broad acceptance of Interactive Systems.

In summary then, the assertion is: Interactive Business Systems are not extensions of earlier business processing techniques, but are a new world of wholly new services not previously available, awaiting your discovery.

INFORMATION INSTINCTS

1. ALL DATA ASSEMBLED IN MY LIFETIME SHOULD BE AVAILABLE INSTANTLY. IN MY BUSINESS, ALL COMPANY HISTORY SHOULD BE AVAILABLE INSTANTLY.
2. ALL INFORMATION IN ALL LIBRARIES OF THE WORLD SHOULD BE AVAILABLE INSTANTLY UPON REQUEST LIKE A WRISTWATCH OR WALLET.
3. INFORMATION SHOULD BE THOUGHTLESSLY REARRANGABLE, WHETHER ALPHABETIC, NUMERIC PHONETIC-ORAL, VISUAL-GRAPHIC, MOTION-KINESTHETIC, ETC.
4. I WANT TO MANIPULATE DATA FREE OF CHARGE. IN THE INFORMATION AGE, WITH INCESSANT DEMANDS FOR INFORMATION REPORTS, THIS IS A NECESSARY FREEDOM OR BIRTHRIGHT. IT SHOULD NOT BE VENDED ON THE BASIS OF WEALTH OR SOCIAL CLASS.
5. FEW IF ANY RULES SHOULD GOVERN THE USE OF INFORMATION SYSTEMS. LIKE THE TV OR PHONE, I DON'T WANT TO KNOW HOW IT WORKS. WITH NO MORE THAN 5 KNOBS, LET ME USE IT.

THE PUBLIC DEMAND FOR INFORMATION SERVICES IS TOO UNCONSCIOUS AND FRANTIC; THE TREND TOO WELL ESTABLISHED TO IMAGINE A FUTURE DISENCHANTMENT WITH THESE IDEALS.

EXHIBIT 8

AN
INFORMATION SYSTEM
FOR
ENROLLMENT PROJECTIONS

ROBERT D. NELSON
Richfield Public Schools
Richfield, Minnesota

INTRODUCTION

School districts make enrollment projections for the short range purpose of planning for next year, which includes allocating staff and students to buildings. Whether to build or to close facilities requires long range enrollment projections covering a five, ten or twenty year period.

Generally, the best predictor of next years enrollment is the trend of the last few years enrollment. The cohort survival ratio technique is appropriate to use if the recent trends affecting enrollment can be assumed to continue. Some of these trends that must remain stable are in-out migration, parochial school attendance, drop out rate and the construction (or demolition) of housing units. Note that the trend is what needs to remain stable or predictable; the data can represent a continuing increase, decrease or even an abrupt change if anticipated.

Enrollment projections based on the last few years enrollment should not be continued more than a few years into the future, five at the very most, unless your school district is very stable.

Long range enrollment projections should be based on factors other than enrollment history. Land use plans, job supply, area birth rates, parochial school plans and socio-economic levels as they relate to family size all affect future school enrollment.

Enrollment projections should be based on good data. A well defined and well maintained data base is necessary to supply the differing types of raw information that short and long range enrollment projections require.

SHORT RANGE ENROLLMENT PROJECTIONS

Assuming relatively stable trends that allow using past enrollment to predict future enrollment, the cohort survival ratio technique can be used. The basic data necessary is the enrollment for the past few years. This data should be gathered for whatever date the enrollment projections are intended. If it is necessary to project enrollment for September 15th, then data on enrollment should be gathered on September 15th of each year.

In order to project the entering kindergarten or first grade, either a family file must be maintained, a guestimate of four or five year olds must be done, or a roundup for eligible children should be held. If projections on the entering grade are done for three or five years into the future, then data on two year olds or newborns needs to be gathered. This data should also be gathered on the projection date (ie, September 15th).

```
                A HYPOTHETICAL SCHOOL DISTRICT

AGE OR   * PAST CENSUS * ENROLLMENT DATA * FUTURE ENROLLMENT PROJECTIONS

GRADE    * 67-68 68-69 69-70 70-71 71-72 * 72-73 73-74 74-75 75-76 76-77
-------------------------------------------------------------------------
AGE 0    *   686   649   608   586   562 *

AGE 1    *   725   688   658   641   618 *

AGE 2    *   771   751   700   683   658 *

AGE 3    *   774   753   736   695   702 *

AGE 4    *   816   725   739   731   704 *
--------------------------------------------
0 - 4    *  3772  3633  3449  3336  3244 *
-------------------------------------------------------------------------
KIND     *   819   768   763   717   719 *   683   683   638   610   585
-------------------------------------------------------------------------
GR  1    *   595   606   572   595   564 *   554   526   506   492   476

GR  2    *   650   588   579   569   550 *   540   530   523   503   471

GR  3    *   604   606   571   590   552 *   539   529   520   493   493
-------------------------------------------------------------------------
1 - 3    *  1849  1802  1722  1754  1666 *  1633  1585  1549  1488  1440
-------------------------------------------------------------------------
GR  4    *   598   589   615   574   585 *   551   538   528   519   492

GR  5    *   607   585   584   609   578 *   582   548   535   525   516

GR  6    *   576   590   582   595   592 *   572   576   542   530   520
-------------------------------------------------------------------------
4 - 6    *  1781  1764  1781  1778  1755 *  1705  1662  1605  1574  1528
-------------------------------------------------------------------------
K - 6    *  4449  4332  4266  4249  4140 *  4021  3930  3792  3680  3553
-------------------------------------------------------------------------
GR  7    *   555   582   618   607   603 *   609   588   592   557   545

GR  8    *   569   550   576   597   613 *   597   603   586   586   551

GR  9    *   589   602   588   642   647 *   666   649   651   632   637
-------------------------------------------------------------------------
7 - 9    *  1713  1734  1782  1846  1863 *  1872  1840  1829  1775  1733
-------------------------------------------------------------------------
GR 10    *   602   585   632   588   647 *   655   674   657   663   639

GR 11    *   540   611   571   599   590 *   635   643   662   645   651

GR 12    *   470   519   581   554   533 *   570   613   621   639   623
-------------------------------------------------------------------------
10 - 12  *  1612  1715  1784  1741  1770 *  1860  1930  1940  1947  1913
-------------------------------------------------------------------------
7 - 12   *  3325  3449  3566  3587  3633 *  3732  3770  3769  3722  3646
-------------------------------------------------------------------------
K - 12   *  7774  7781  7832  7836  7773 *  7753  7700  7561  7402  7199
-------------------------------------------------------------------------
```

Figure 1

COHORT SURVIVAL RATIO TECHNIQUE

The following data (from figure 2) is used to illustrate the calculations necessary in utilizing the cohort survival ratio method.

	1967-68		1968-69		1969-70		1970-71		1971-72
GRADE 3	604		606		571		590		552
		0.9752		1.0149		1.0053		0.9915	
GRADE 4	598		589		615		574		585

A survival ratio is, for example, the number of students in the 1968-69 grade 4 divided by the number of students in the 1967-68 grade 3. (Numerically, 589/604 = 0.9752) Sometimes, the survival ratios are expressed as percentages (0.9752 would be the -2.48% as shown on figure 2).

The four survival ratios calculated if five years of data are used determine a future survival ratio for that age or grade level. The future survival ratio can be calculated as a simple average or utilizing a weighted average. Using weights of 1,2,3, and 4 to emphasize the most recent data:

$$\frac{1 \times .9752 + 2 \times 1.0149 + 3 \times 1.0053 + 4 \times .9915}{1 + 2 + 3 + 4} =$$

$$\frac{.9752 + 2.0297 + 3.0158 + 3.9661}{10} = \frac{9.9867}{10} = 0.9987$$

```
                    A HYPOTHETICAL SCHOOL DISTRICT

AGE   * PAST CENSUS * ENROLLMENT DATA * YR TO YR % CHANGE * PROJECTIONS
OR
GRADE * 67-68       68-69       69-70       70-71       71-72 *        72-73

AGE 0 *   686         649         608         586         562 *
      *        -5.39%      -6.32%      -3.62%      -4.10%     *
      *        +0.29%      +1.39%      +5.43%      +5.46%     *  +4.12%
AGE 1 *   725         688         656         641         618 *          585
      *        -5.10%      -4.36%      -2.58%      -3.59%     *  -5.34%
      *        +3.59%      +2.91%      +3.80%      +2.65%     *  +3.14%
AGE 2 *   771         751         708         683         658 *          627
      *        -2.59%      -5.73%      -3.53%      -3.66%     *  -3.19%
      *        -2.72%      -2.00%      -1.84%      +2.78%     *  -0.11%
AGE 3 *   774         750         736         695         702 *          657
      *        -3.10%      -1.87%      -5.57%      +1.01%     *  -6.41%
      *        +2.71%      -1.47%      -0.68%      +1.29%     *  +0.29%
AGE 4 *   816         795         739         731         704 *          704
      *        -2.57%      -7.04%      -1.08%      -3.69%     *  +0.02%
      *        -5.88%      -4.03%      -2.98%      -1.64%     *  -2.94%
KIND  *   819         768         763         717         719 *          683
      *        -6.23%      -0.65%      -6.03%      +0.28%     *  -5.01%
      *       -26.01%     -25.52%     -22.02%     -21.34%     * -22.85%
GR  1 *   595         606         572         595         564 *          554
      *        +1.85%      -5.61%      +4.02%      -5.21%     *  -1.77%
      *        -1.18%      -4.46%      -0.52%      -7.56%     *  -4.17%
GR  2 *   650         588         579         569         550 *          540
      *        -9.54%      -1.53%      -1.73%      -3.34%     *  -1.82%
      *        -6.77%      -2.89%      +1.90%      -2.99%     *  -1.83%
GR  3 *   604         606         571         590         552 *          539
      *        +0.33%      -5.78%      +3.33%      -6.44%     *  -2.36%
      *        -2.48%      +1.49%      +0.53%      -0.85%     *  -0.13%
GR  4 *   598         589         615         574         585 *          551
      *        -1.51%      +4.41%      -6.67%       1.92%     *  -5.81%
      *        -2.17%      -0.85%      -0.98%      +0.70%     *  -0.42%
GR  5 *   607         585         584         609         578 *          582
      *        -3.62%      -0.17%      +4.28%      -5.09%     *  +0.69%
      *        -2.80%      -0.51%      +1.88%      -2.79%     *  -0.93%
GR  6 *   576         590         582         595         592 *          572
      *        +2.43%      -1.36%      +2.23%      -0.50%     *  -3.37%
      *        +1.04%      +4.75%      +4.30%      +1.34%     *  +2.88%
GR  7 *   555         582         618         607         603 *          609
      *        +4.86%      +6.19%      -1.78%      -0.66%     *  +1.00%
      *        -0.90%      -1.03%      -3.40%      +0.99%     *  -0.99%
GR  8 *   569         550         576         597         613 *          597
      *        -3.34%      +4.73%      +3.65%      +2.68%     *  -2.61%
      *        +5.80%      +6.91%     +11.46%      +8.38%     *  +8.75%
GR  9 *   589         602         588         642         647 *          666
      *        +2.21%      -2.33%      +9.18%      +0.78%     *  +2.94%
      *        -0.65%      +4.98%      +0.00%      +0.78%     *  +1.24%
GR 10 *   602         585         632         588         647 *          655
      *        -2.82%      +8.03%      -6.96%     +10.03%     *  +1.24%
      *        +1.50%      -2.39%      -5.22%      +0.34%     *  -1.76%
GR 11 *   540         611         571         599         590 *          635
      *       +13.15%      -6.55%      +4.90%      -1.50%     *  +7.63%
      *        -3.89%      -4.91%      -2.98%      -2.67%     *  -3.39%
GR 12 *   442         519         581         554         583 *          570
      *       +17.43%     +11.95%      -4.65%      +5.23%     *  -2.23%
```

Figure 2

This future survival ratio calculated for each age or grade level is held constant and used to project the enrollments for the next five years. (A 1971-72 grade 3 enrollment of 552 multiplied by the grade 3 to grade 4 future survival ratio of 0.9987 would yield 551 for the projected 1972-73 grade 4 enrollment).

REPORTS

An ENROLLMENT PROJECTION report (figure 1) should show the last few years of census and enrollment data and the next few years of projections. This enables the reader to identify the recent trends and heuristically confirm the extrapolation of those trends in the projections.

A YEAR TO YEAR CHANGE report (figure 2) includes the percent changes in an age or grade level from one year to the next (the top percentage, ie, the 1967-68 to 1968-69 age 0 change from 686 to 649 was a decline of 5.39%) as well as the percent changes that occur as a class ages (the bottom percentage, ie, age 0 to age 1 the next year changes from 686 to 688 for an increase of 0.29%). These percentages give a feel for trends and the magnitude of change over a period of time.

A HYPOTHETICAL SCHOOL DISTRICT

ACTUAL VS PROJECTED ENROLLMENTS FOR 1973
BASED ON DATA FROM 1967 THROUGH 1971

		ACTUAL ENROLL	PROJ'D ENROLL	ACTUAL - PROJ	% ERROR (A-P)/A
KIND	*	671	683	-12	-1.79
GR 1	*	541	526	+15	+2.77
GR 2	*	536	530	+6	+1.12
GR 3	*	537	529	+8	+1.49
1 - 3	*	1614	1585	+29	+1.80
GR 4	*	535	538	-3	-0.56
GR 5	*	548	548	+0	+0.00
GR 6	*	562	576	-14	-2.49
4 - 6	*	1645	1662	-17	-1.03
K - 6	*	3930	3930	+0	+0.00
GR 7	*	574	588	-14	-2.44
GR 8	*	613	603	+10	+1.63
GR 9	*	660	649	+11	+1.67
7 - 9	*	1847	1840	+7	+0.38
GR 10	*	655	674	-19	-2.90
GR 11	*	650	643	+7	+1.08
GR 12	*	603	613	-10	-1.66
10 - 12	*	1908	1930	-22	-1.15
7 - 12	*	3755	3770	-15	-0.40
K - 12	*	7685	7700	-15	-0.20

Figure 3

A COMPARISON report (figure 3) is useful to illustrate the past accuracy of whatever method is used. Projections made for some past year are compared with the actual enrollment for that year and the differences are noted. Comparison reports for a number of past years can be convincing either that the enrollment projection method seems to be applicable to the school district or that it should not be used.

Various GRAPHS which show the enrollment by levels (ie, elementary, junior high, total, etc.) can also show the public school enrollment compared to the total number of children residing in the district (figure 4). A chart can also follow a particular class through school and portray in or out migration and parochial school attendance patterns (figure 5).

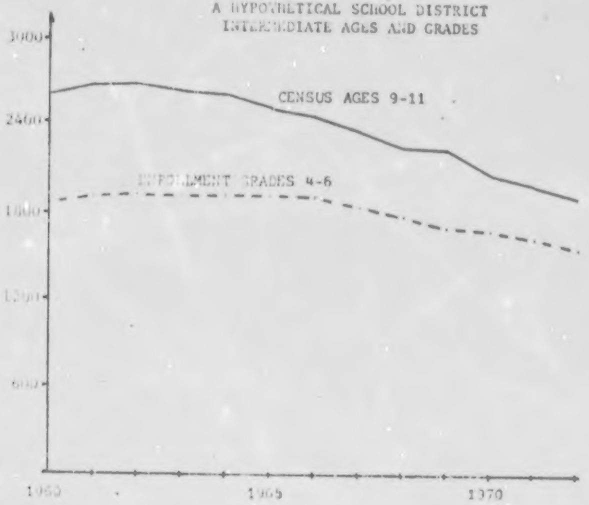


Figure 4

LONG RANGE ENROLLMENT PROJECTIONS

Current trends which can change drastically within just a few years time make long range projections very risky. The factors affecting school district enrollments need to be identified and followed.

A DWELLING YIELD study identifies the average number of adults, children or students that each type of dwelling unit (ie, single family home, 2 BR apartment, mobile home, etc.) presently holds. These ratios can be used to project the yield from any new construction or construction made probable by zoning changes. If each single family home presently yields 1.6 children ages 5 to 16, then 100 similar new homes would have approximately 160 children ages 5 to 16.

A POPULATION TREE is a graphic device to show the age and sex structure of a school district. Areas in different stages of development have characteristic shapes such as the new suburb filled with young families and small children (figure 6) and a mature suburb with declining enrollment (figure 7). Population trees for each dwelling type or sub area of a school district are useful.

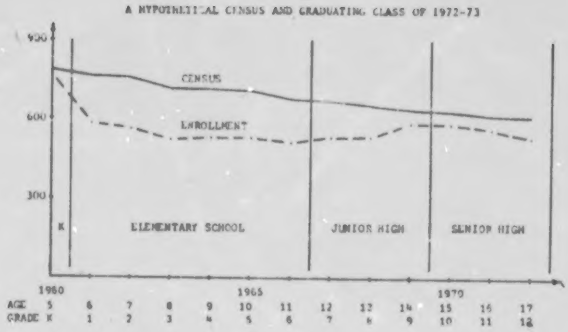


Figure 5

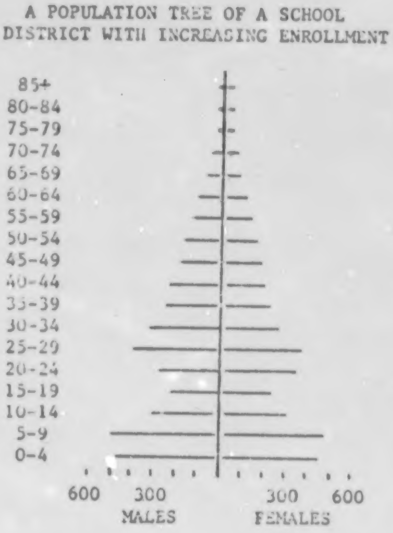


Figure 6

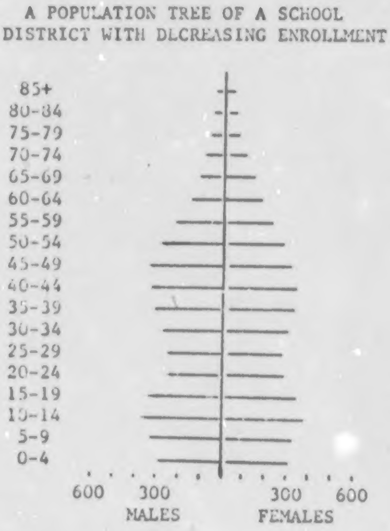


Figure 7

Information about SOCIO ECONOMIC CHARACTERISTICS such as BIRTH RATES and FAMILY SIZE help in projecting enrollment. Applying birth rates to the number of women of child bearing age or family size to the number of young marrieds will yield some estimates, although these ratios are currently in a very unstable situation.

A CHILDREN TO POPULATION RATIO can sometimes be used to project a peak enrollment or the bottom of a decline. Many new communities have a 1:2 ratio of children to total population, but twenty years later this ratio may be 1:4 (figure 8). If total population projections are available and a ratio can be derived from older communities, a projected child population can be roughly estimated.

A HYPOTHETICAL SCHOOL DISTRICT
TOTAL POPULATION, AGES 0-16 and 0-5

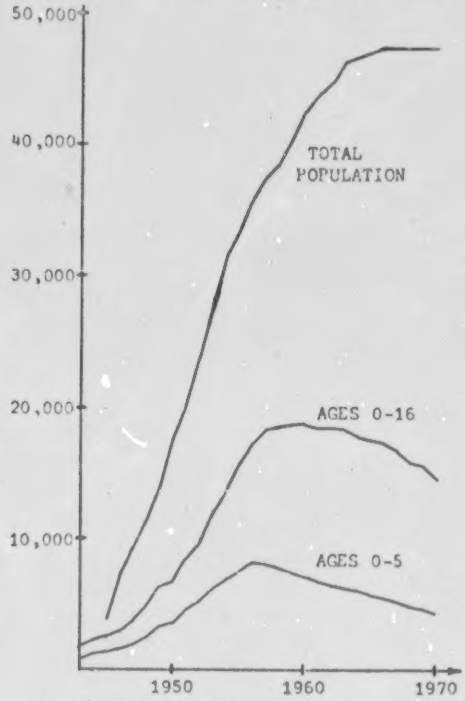


Figure 8

MIGRATION patterns can be followed to analyze enrollment trends. For example, an identifiable out migration of families living in apartments whose oldest child is in the lower elementary grades should support a high preschool population in apartments not having an effect on the enrollment in upper elementary grades or secondary schools. A knowledge of the amount of in or out migration will help to assess the impact that easy or tight mortgage money would have on the school district enrollment. Moves within the district need to be separated so in and out migration rates are not inflated.

The impact of FREEWAYS, RETAIL BUSINESSES or INDUSTRIAL CENTERS can result in the opening of new areas for population growth or cause a drop as dwelling units are removed during construction.

INFORMATION SYSTEMS FOR ENROLLMENT PROJECTIONS

Enrollment projections are generally made by analyzing data from reports that are generated year after year at the same time of the year. The information system should contain the data items necessary for the reports and the capability of retrieving the data to generate the reports.

A STUDENT FILE should contain each students birthdate, school, grade level, entry, withdrawal and reentry codes and dates and a reference to the student's family.

The FAMILY FILE should have the parent's birthyears, other children in the family and their birthdates, the total number of children, the month and year the family moved into their present dwelling as well as when they moved into the school district, the previous residence, the new residence (when leaving the school district) and a reference to the current residence.

The RESIDENCE FILE should have the dwelling type, the sub area of the school district, the year of construction, and possibly turnover indicators such as the number of families that have lived in that residence, the year each moved in and out, and the birthyears of the occupants.

DATA BASE ITEMS FOR ENROLLMENT PROJECTIONS

STUDENT

- School
- Grade
- Birthdate
- Entry, withdrawal and reentry codes and dates

FAMILY

- No. of children
- Birthdate of each child
- Public and nonpublic school attendance pattern
- Birthyear of head of house and spouse
- Mo/yr moved into school district
- Mo/yr moved into present dwelling
- Previous residence
- New residence - when leaving school district

RESIDENCE

- No of families
- For each family that has lived in the dwelling
  - year moved in
  - year moved out
  - birthyears of all family members
- Sub area of the district
- Dwelling type
- Year constructed

SUB AREAS

The district sub areas should be well conceived and not changed unless subdivided. Do not use elementary school areas since these boundaries are subject to change, especially when a school is closed or a new school constructed.

Enrollment projections for elementary schools within a school district are generally difficult due to the small numbers and the fluctuating boundaries. In some cases, it is possible to group sub areas with similar characteristics - even if not contiguous - and determine indicators such as dwelling yields, survival ratios, etc, for the group of sub areas. Then these indicators can be applied to the data in each sub area. Finally the projections for the sub areas that comprise the particular area of interest can be aggregated to arrive at projections for schools within the school district. If school boundaries change, it is only necessary to combine different sub areas to determine the appropriate enrollment projections.

If this methcd of grouping sub areas to determine indicators, applying the indicators to individual sub areas, and aggregating to arrive at enrollment projections for school areas, municipalities, etc. is utilized, then the sub areas need to be small and quite homogeneous with respect to the indicators used. A sub area should be five to ten square blocks, 100 to 200 single family homes, or possibly an apartment complex.

DISTRICT AND SUB AREA ANNUAL REPORTS

Many reports should be generated for each sub area and for the district each year and saved for future reference. The basic information is population by age and sex; students by grade; number of dwellings by type; dwelling yields showing the population by age and dwelling type, and students by grade and dwelling type; migration studies containing number of families, population and student counts by head of house, age and year moved in as well as year moved out, and the extent of in district migration.

Data and reports from other sources should be included. The Federal Census is an excellent source of socio economic data. Zoning changes and construction starts should be monitored. Non public school attendance should be documented. Area birth rates and migration indicators such as availability of mortgage money can be gathered.

CONCLUSION

Accurate enrollment projections require good data, analysis and luck. The basic indicators are past enrollment trends, migration, birth rates and dwelling yields. As long as these indicators behave and/or remain predictable, good enrollment projections are possible. Other data collection and analysis is necessary to keep tabs on the indicators. Consistent data collection and reporting permits variations in the trends to be identified early. Appropriate analysis is then used to translate these changes into revised enrollment projections.

COMPUTER-AUGMENTED TEACHER TRAINING:

AN UNDERGRADUATE PROGRAM IN MATHEMATICS TEACHER EDUCATION

by

Portia C. Elliott and Howard A. Peelle

University of Massachusetts

INTRODUCTION

"Eventually, programming itself will become more important even than mathematics in early education." -- Marvin Minsky (13)

The preceding quotation suggests a dramatic departure from the proverbial three R's in elementary education and prophesies the inclusion of a powerful "P"--programming--in the curriculum of tomorrow. Programming is not to be confused with programmed instruction (PI) or narrowly construed uses of computer-assisted instruction (CAI) which places students in passive, almost docile roles by giving them prepackaged programs with preordained answers and prescribed paths to those answers. Rather, programming (in an education context) involves students actively directing the computer to solve problems or to create original projects. Programming is fundamentally a learner-centered activity--one which allows the learner to take increasing responsibility for his own educational destiny.

Recently, other computer scientists and educators have envisioned programming in the elementary school classroom. Dwyer (4), Luehrmann (11), Milner (12), Papert (14), and Peelle (16), all have advocated student-controlled computing and non-exploitative uses of computer technology in education. Papert's (14 (p.   view of an ideal relationship between student and computer is particularly refreshing:

> . . . technology is used not in the form of machines for processing children, but as something the child himself will learn to manipulate, to extend, to apply to projects, thereby gaining a greater and more articulate mastery of the world; a sense of power or applied knowledge and a self-confidently realistic image of himself as an intellectual agent.

Many of the activities and concepts involved in programming a computer are, indeed, important to learn. The notions of "bug" and "debugging"; of algorithms (procedures) and sub-procedures; of iteration and recursion; and of feedback and heuristics are intellectually "powerful ideas" (to borrow again from Papert). They are ideas we rely upon every day--perhaps unconsciously--to process information from the world we perceive and to make decisions, plan, think, etc. In brief, these ideas are important tools which can serve as building blocks for further cognitive development.[1]

The question then arises: "What about teachers and their development?" What knowledge and what skills should teachers have when computer programming is accepted in elementary schools? The answer is not found in the simplistic suggestion: "Take a computer course." Most (so-called) computer courses are not designed with applications for teaching in mind. And yet, if concepts of programming are valuable cognitive tools for students to have, are they not for their teachers as well?

Unfortunately, teacher training in computer usage has been mostly neglected. In their report, "Factors Inhibiting the Uses of Computers in Education," Anastasio and Morgan (1) cited lack of adequate teacher training programs as a major deterrent to computer use in schools. The Conference Board of the Mathematical Sciences' report (2) listed the following as one of their recommendations:

> We recommend . . . the development of a variety of programs for the training of teachers and of teachers of teachers of high school courses involving computers.

But, teachers themselves are notorious for resisting innovation. They seem to harbor a lot of myths and fears about technology--particularly about computers. For some teachers, the computer is seen as a vulgar intrusion into their private domain--the classroom--and threatens their very jobs; for others, computers are wonder-machines which "can do anything" or (worse yet) are "smarter than people"; and for many, there is a deep reluctance to use sophisticated equipment--usually expressed as "No thank you . . . I might break something" in order to avoid exhibiting ignorance. These fears and myths are, of course, largely based on misunderstandings or lack of understanding about computers.[2]

So, the problem for teacher education remains: *How* should we train teachers who will be exposed to computers and programming? Granted, not all teachers can teach programming effectively; some may become expert in other aspects of computer usage, such as curriculum development. But at a minimum, teachers need to be able to convey the importance of knowing about computers and be able to arrange computing environments which foster the development of competent learners.

COMPUTER-AUGMENTED TEACHER TRAINING--AN OVERVIEW

A new teacher education program has been inaugurated at the University of Massachusetts' School of Education under the title: "Computer-Augmented Teacher Training" (abbreviated CATT). CATT is an undergraduate program designed to train prospective teachers of mathematics.[3]

The primary goal of the CATT program is to develop teachers' competencies, utilizing the computer where and when it augments their training. Three sub-goals of teaching competency are identified as follows:

1. Developing Cognitive Competencies

Development of competence in mathematics content areas (what to teach); and development of competence in methods of effective teaching (how to teach).

2. Developing Affective Competencies

Development of self-concept (confidence in teaching) expression and management of emotions (self-control).

3. Developing Social Consciousness

Learning the advantages and limitations of technology (what computers can and cannot do); understanding man-machine interaction (computer literacy); awareness of critical issues in a technological society.

COGNITIVE COMPETENCE: KNOWING WHAT TO TEACH AND HOW TO TEACH

In order to strengthen teachers' cognitive competencies, the CATT program offers training in both content and methodology; this is, prospective teachers are taught *what* to teach and *how* to teach. Trainees develop their understanding of mathematics as they are re-introduced to fundamental mathematical concepts via a programming language. (See "Learning Mathematics Via A Programming Language" below.) Trainees also become acquainted with pedagogical strategies for teaching mathematics effectively, emphasizing use of computer technology and programming paradigms. (See "Teaching Mathematics Via A Programming Language" below.)

Learning Mathematics Via A Programming Language

Based on the idea that "a good way to learn something is to teach it," teachers-in-training in the CATT program "teach" the computer. That is, they learn mathematics by communicating their newly assimilated knowledge to a computer, using A Programming Language.[4]

The role of the computer is like that of a "model student." The computer responds only to explicit instructions; it does exactly what it is told (even if that is different from what one thought one told it); once it is "taught" something (like a rule), it does not forget or distort it and can apply it accurately upon command; and it is (usually) ready to accept more instruction, tirelessly and obediently.[5]

Thomas Kurtz (10) and the Dartmouth Secondary School Project staff are proponents of this "computer-as-pupil" approach for high school students:

> Because we had to *teach* an ignorant machine, we were *forced* to break the process down into pieces, arrange these pieces in the proper order, and present them to our pupil machine and see if our instructions were presented in a logical, foolproof way that it could follow. Before we made the effort to teach this pupil, we were forced to clearly understand the problem ourselves.

The approach applies equally well to teacher training.

Teaching Mathematics Via A Programming Language

In this aspect of their training, prospective teachers become acquainted with strategies for teaching mathematics effectively. This includes the pedagogy of using the computer directly in the teaching process. Not only do these teachers learn about computer use and its application in mathematics, but also they are shown how computer programming, in particular, can be used as a *means* for teaching mathematics.

Why use computer programming in training teachers? Because certain processes intrinsic to programming a computer are analogous to some powerful methods of teaching. For instance, only by programming can prospective teachers learn to *control* the computer and "debug" programs (eliminate errors). Just as a teacher must learn how to handle academic and disciplinary "bugs" in the classroom, she/he can find strategies which work for "debugging" computer programs.

In the role of "model student" (described earlier), the computer allows one to do some "practice teaching"--with no detriment to real students! The teacher treats the computer as a simulated student--with potential for developing its innate abilities --and many observe the effects of different teaching approaches.

A variety of pedagogical strategies to the prospective teacher. The strategies are actually a set of heuristics which teachers may use to facilitate student learning activities.6 For example, they might:

--Use Manipulative Materials
--Encourage Articulation of Thinking Processes
--Encourage Articulation of Emotional Reactions
--Allow Increasing Learner Control
--Change Pedagogical Strategies

AFFECTIVE COMPETENCE: SELF-CONTROL AND CONFIDENCE IN TEACHING

In order to develop the affective competencies of teachers, they are placed in active learning environments--environments in which both success and failure are likely to occur. (This is characteristic of the type of environment in which they will be eventually teaching.) That is, by actively dealing with their own emotional reactions to success and failure, prospective teachers can improve their self-control and develop confidence in their teaching ability.

Once again, the programming paradigm helps. When a teacher-in-training writes a program, it either works properly or it doesn't. Success or failure is easily recognized. By contrast, in human affairs, success and failure are often ambiguous and difficult to identify. When a program is the object of attention, there can be joy in seeing that "it works!"; or, of course, there can be frustration caused by irrepressible "bugs". But, in either case, an emotional reaction is illicited; and that reaction is explicit and focused--but not at another person--making it easier to deal with.

Learning to manage one's emotional reactions can be facilitated by the presence of a computer. When a programmer is successful, she/he has a medium available for automatically and proudly displaying results. When a programmer is not successful, she/he can debug the program in private consultation with the computer and work through the accompanying emotional stress without any additional social embarrassment. The motivation for dealing with failure constructively is strong--for things will certainly not get any better unless you do something about it. After all, the computer cannot fix your program!

SOCIAL CONSCIOUSNESS: COMPUTER LITERACY

In order for teachers to become more aware of the extensive commitment our society has made to technology, the CATT program is designed to develop their social consciousness. Cognitive and affective teaching competencies are not enough. Teachers need to know about the advantages and limitations of advanced technology--in particular, what computers can and cannot do.

Developing such a "computer literacy" is important if a teacher is to effectively deal with problems of a technologically-oriented society. Several studies of computer use in education have cited computer literacy as a high priority.7 Accordingly, the CATT program prepares prospective teachers with special skills, such as ability to:

--Identify cases when technology is used as a shield behind which decision-makers hide, e.g., "The computer gave us this answer" (school Busing route);

--Critically analyze curriculum materials for negative racial or sexist overtones, particularly computer-based curriculum;

--Debunk myths (e.g., computer omnipotence) and assuage fears (e.g., "The computers are after our jobs").

These skills translate back to computer literacy. For, as Alvin Tofler (17) put it: "Tomorrow's schools must . . . teach not merely data, but ways to manipulate it." The computer is man's most powerful and versatile tool for manipulating data. Indeed, it is a supreme symbol manipulator--performing instructions at speeds measured in nanoseconds, which far exceed man's own capabilities.8

But, the major reason for including computer literacy as a significant component of a teacher education program is not so that teachers (or anyone else) will compete with computers. Rather, teachers should be at the forefront, charged with the responsibility for teaching computer literacy (along with other essential literacies). They themselves must, therefore, learn how and why computers are used. Specifically, this involves:

--Knowledge of computers (what a computer is; how a computer works; etc.);

--Knowledge of computing (information processing; algorithms, etc.);

--Knowledge of applications (what computers can do/can't do);

- --Experience in computer usage (man-made interaction);
- --Using computers in teaching (computer-assisted instruction; computer-managed instruction);
- --Programming (how to control the computer);
- --Issues and Implications of Computer Ubiquity (data banks vs. privacy; artificial intelligence, etc.).

CONCLUSION

In summary, the CATT program described here has several distinguishing features which make it unique in teacher training. First, the computer--and A Programming Language in particular--play a central role in the development of prospective mathematics teachers. Second, modern technology is interwoven with traditional teaching approaches and content is blended with methods to make a pal table mixture for teacher trainees. Finally, the program stresses development of cognitive, affective and social competencies in prospective teachers to ensure that future learners will have teachers who are more knowledgeable, more sensitive, and more humane.

NOTES

[1] Some preliminary research efforts have attempted to connect Piaget's developmental theories to the learning of programming concepts. See dissertations by Michael Folk, "Influences of Developmental Level on a Child's Ability to Learn Concepts of Computer Programming," (Syracuse University), 1972; and Joyce Statz, "The Development of Computer Programming Concepts and Problem-Solving Abilities," 1973, (Syracuse University).

[2] For further explication of this point, see Bork (3) and Peelle (15).

[3] While this program is open to prospective teachers of mathematics at all levels --elementary, intermediate, and secondary--it is described here only in terms of elementary mathematics teacher training. (A complete description may be obtained from the authors.)

[4] APL (A Programming Language) is a general-purpose interactive computer programming language developed by Kenneth Iverson (7) and supported by IBM. Originally conceived as a unifying mathematical notation, APL is a language with simple rules and yet offers a user great computational power and flexibility. APL has been applied successfully in fields such as business, scientific research, and education. (See Iverson (8).)

[5] If a computer is not available, the activity of "dry" programming is still valuable for cognitive development. See Iverson (9) for a description of using A Programming Language as a notation for clarifying concepts and procedures.

[6] These are "heuristics" in the sense that they are not guaranteed to work for all students at all times. See Dwyer (4) for a broader description of "Heuristics for Using Computers to Enrich Education."

[7] See, for example, Begle (2).

[8] In the area of pattern recognition, however, the computer does not fare as well, comparatively. A major area of research in artificial intelligence is devoted to what humans regard as relatively simple tasks of pattern recognition, such as distinguishing between (script) letters of the alphabet.

REFERENCES

(1) Anastasio, E. and Morgan, J. National Science Foundation Report: "Factors Inhibiting the Uses of Computers in Education," 1972.

(2) Begle, Edward et al. Report of the Conference Board of the Mathematical Sciences Committee on Computer Education: "Recommendations Regarding Computers in High School Education," April, 1972.

(3) Bork, Alfred. "The Computer in Teaching--Ten Widely Believed Myths," ACM SIGCUE Bulletin, Volume 7, Number 4, October, 1973.

(4) Dwyer, Thomas A. "Heuristic Strategies for Using Computers to Enhance Education," International Journal of Man-Machine Studies, 1974.

(5) Elliott, Portia C. "Elementary Mathematics Teacher Training Via A Programming Language." Unpublished Ed.D. dissertation, University of Massachusetts, 1973.

(6) Furth, Hans and Wachs, Harry. Piaget's Theory in Practice: Thinking Goes to School. New York: Oxford University Press, 1974.

(7) Iverson, Kenneth E. A Programming Language. New York: John Wiley and Sons, Inc., 1962.

(8) Iverson, Kenneth E. "APL in Exposition," IBM Technical Report, Number 320-3010, Philadelphia Scientific Center, January, 1972.

(9) Iverson, Kenneth E. "APL As An Analytic Notation," Proceedings of APL V Conference, Toronto, Canada, May 1973.

(10) Kurtz, Thomas E. "The Computer As Pupil: The Dartmouth Secondary School Project," NSF GW-2246, *Final Report*, October, 1970.

(11) Luehrmann, Arthur W. "Should the Computer Teach the Student or Vice-Versa?", *AFIPS Conference Proceedings*, Volume 40.

(12) Milner, Stuart. "Learner-Controlled Computing: A Description and Rationale," Paper presented at the Annual Meeting of the American Educational Research Association, Chicago, Illinois, April, 1974.

(13) Minsky, Marvin. "Form and Content in Computer Science," *Journal of the ACM*, Volume 17, Number 2.

(14) Papert, Seymour. "Teaching Children Thinking," *MIT Artificial Intelligence Memorandum*, Number 247.

(15) Peelle, Howard A. "Pygmalion's Computer," in *Controversies in Education*, edited by Allen and Hecht. Saunders, Inc., 1974.

(16) Peelle, Howard A. "Computer Glass Boxes: Teaching Children Concepts With A Programming Language," *Educational Technology*, Volume XIV, Number 4, April, 1974.

(17) Tofler, Alvin. *Future Shock*. New York: Random House, 1970.

INTERACTIVE REPEATABLE TESTING:
A MANY--FACETED TOOL

David P. Yens and Victor R. Martuza
University of Delaware

Many college and public school teachers have, within the last few years, found that the traditional lecture and test method of instruction does not meet the needs of their students for adequate mastery of course materials and interest in course content. This has led to a search for new instructional methods and has resulted in the implementation of open classrooms, independent study, modularized instruction, performance or competency based instruction, and other non-traditional methods. A necessary ingredient for the success of several of these methods is an effective testing program which incorporates the development and administration of test items and the analysis of results. Several people across the country have recently started to use the computer for help in the testing process because of its storage capacity, ability to manipulate large data bases, and analytical capabilities. Most test-related uses of the computer have been oriented toward test scoring and the development and evaluation of large pools of items. For example, tests might be printed by the computer and copied for the students who respond using mark sense forms. These forms are fed into the computer with item identification information and the resulting data may be stored and used for later analyses. An excellent review of several computer assisted test construction projects is provided in the March, 1973 issue of *Educational Technology*.

Origins of the Computer Assisted Testing Project

At the the University of Delaware, several faculty in the College of Education have developed courses that are partially or completely designed to permit independent study. They were especially concerned with ways of permitting students to demonstrate mastery of each curriculum module or unit. It was believed that the learning process could be improved by permitting a student more than one attempt at a criterion test, but this would require several different copies of a test for each unit. To solve this problem, the faculty turned to the computer.

In 1969 a computer program was developed by Teresa Green to generate randomly parallel tests with a specified number of questions from a large pool of test questions. Test questions were randomly selected so that any item might appear on any copy of the test, but the same item would not appear more than once on a particular test. A blank was provided for the student to enter his response to each question, and a separate strip of the output was reserved for the answers. This answer strip could be separated from the printout containing the questions before the test was given to the student. An instructor could then compare the strip with the student responses and easily grade the test. The randomization of the questions made it possible for a student to take a test two or three times with a low probability of seeing a question repeated. Note that each student had his own copy of the test printout; tests did not need to be reproduced in the usual way.

The basis of this competency-based modular learning process was carefully specified behavioral objectives to guide a student's study of assigned reading materials. A student's grade depended upon the number of points he accumulated for his work. The tests were criterion-referenced in that a pre-specified minimum score had to be achieved before credit could be granted for the unit. Clearly, it was advantageous to permit a student to take more than one test on a unit to insure that he had reached an acceptable level of competence.

Based on our experience, the paper and pencil tests have worked reasonably well, but they have several disadvantages. Several hundred tests must be generated periodically and carried to the testing location. Answer strips must be separated from the questions. Someone must always be present to distribute and score the tests and periodically file them. With large numbers of tests and students these tasks are time-consuming, arduous, and likely to result in errors.

As with all paper and pencil tests, students have the opportunity to view all questions at once and may obtain aid from one question in answering another. The students, of course, may see this as an advantage, but it may lead to large differences in test difficulty so that any given test may not provide a true indication of a student's knowledge. This is especially true if several items refer to one concept. For the courses that have been developed thus far, this problem has not been especially noticeable, but the potential for difficulty exists.

When test items are used repeatedly, it is desirable to obtain item analysis data for the evaluation of the items. In this way, items that are especially easy or difficult or which are confusing or misleading can be detected and rewritten as necessary. With paper and pencil tests, troublesome items can be detected through discussions with students, but the psychometric properties of the items cannot be determined because the random nature and the large number of tests makes the application of item analysis techniques quite difficult.

The Computer Assisted Testing Program

A solution to several of the problems with paper and pencil tests, and the next logical step in the testing program, was the development of an administration process in which the computer administers the test directly to the student. An additional reason for taking this step was the desire to introduce education students to computers before they encounter computer facilities in schools in which they will soon teach. Finally, this was seen as an important step toward the establishment of a comprehensive computer assisted instruction and computer managed instruction program which would meet a critical need within the College.

Three cathode ray tube (CRT) terminals were ordered and programming was initiated in December, 1972. The first students took tests via the terminals in April, 1973. Initially, the program was very simple, but extensive improvements have been made, three more terminals have been obtained, and, recently, upwards of 200 tests a day have been administered.

The six terminals are presently connected via 300 baud acoustic couplers to a general purpose Burroughs B6700 computer that handles all the University's computing. The testing center is open 50 hours a week and is staffed by one full time proctor and several part time student proctors who provide assistance to 800 or so students taking the independent study courses currently on the system. Shortly after the start of a term the students generally know what to do, thus freeing the proctors for additional duties such as data tabulation and maintenance of student files. To provide a backup in the event of occasional computer down time or when the student load becomes too great, paper and pencil tests are available and are administered and scored by the proctors when required.

In our system the students' interaction with the terminal was designed to be as simple as possible. All student responses are set up as formatted input; the student answers each question as it is asked. By faculty request, feedback is provided after each response, and a student continues to respond until he selects the correct answer. A comment option is provided that permits the student to store a narrative comment for each question and for the whole test. These comments are incorporated into the student record file for later review by faculty. Upon completion of the test, a summary of the performance is displayed.

The program contains a number of options for controlling the interaction with students and for generating a variety of student records. Items can be sampled from the item pool with or without replacement for specific students and/ or specific tests. Feedback can be modified or eliminated from the testing process if desired.

The data base contains identification, item response, response latency, and comment information for each test administered. Several options for generating student records are available. Both student-oriented and item-oriented outputs are provided, and several parameters may be manipulated to select on different attributes in the file. This student records system provides a powerful means of analyzing item responses and improving the item pools.

A new addition to the program is an automatic records-keeping routine that automatically stores a student's score in a separate file. Thus, all scores for all tests are directly addressable for each student. This file is dumped periodically to permit students to review their progress and to insure accuracy of the results. Of course, at the end of the semester this file provides a complete record of each student's performance. A special utility routine permits the addition of scores from the paper and pencil tests to this student scores file.

Student and Faculty Attitudes toward CAT

Student attitudes toward the use of computers for testing and instruction have been assessed with a 59-item questionnaire (1) for several terms. Attitude information has also been gained through the use of course attitude surveys and through discussion with the students. In addition, performance data for one course have been analysed.

In contrast to normal uses of attitude surveys, we administered the form both before and after testing and were able to assess changes in attitude. The pre-testing results indicate that students have a slightly positive attitude toward the use of computers for testing and instruction, but are concerned about the mechanical aspects such as the ease of operating a terminal.

The first major use of the attitude questionnaire was during the spring semester of 1974. This turned out to be a period of numerous problems and problem solving activities for our programming staff and the Computing Center staff. System crashes and program difficulties occurred more often than were desirable. Consequently, it was not surprising to find a decrease in positive attitude toward computer uses for testing and instruction. However, on the Likert-type items, the means of the responses were typically rather close to the neutral point which, under the circumstances, indicates that the average student will accept the use of computers for testing even if he encounters occasional computer problems. Furthermore, student attitudes were more positive toward the mechanical aspects of the operation; students found the terminal operation easier than anticipated.

---

(1) This questionnaire is an extensive revision of one developed by Bob Brown at the Pennsylvania State University (Brown, 1966).

Despite the difficulties encountered by the students, an independent course evaluation conducted by the instructor of a psychology course (for which 400 students were taking tests via the terminals) indicated that students strongly preferred the testing by terminals to the more traditional testing process, and they thought they learned more than in a "traditional" course. They viewed the computer quizzing as a very valuable aspect of the course. Almost all students in all courses gave high marks to the opportunity to work at their own pace, knowing the results after each question, and being able to take more than one test. As might be expected, there were some students who intensely disliked the use of terminals or were afraid to use them, no matter what the environment.

During the fall of 1974, the computer and programs worked quite well; the environment was about as good as it will get. A mid-term questionnaire was administered to 130 students. Nineteen percent reported they were afraid to use the terminal or hated to use it. However, thirty percent enjoyed using it, 32 percent didn't mind using it, three percent had no opinion, and 16 percent preferred not to use the terminal. Most of the negative group seemed to be concerned about seeing the whole test and being able to go back and change answers. However, questionnaire results indicated that this problem decreased with increasing terminal use.

Typically, on course evaluation questionnaires that permit students to specify most liked and most disliked features of the course, the use of terminals is mentioned about equally in both categories. This seems to be independent of the ability of the student, his total attitude toward the course, and the testing environment.

The analysis of scores for quizzes taken by the same students on terminals and hard copy indicates that there is no marked difference in performance. It appears that the use of terminals makes little difference even though some students feel that it does.

Faculty have made extensive use of the item analyses provided at the end of each term. Corrections, additions, and deletions to the test question file may be easily made using the time-sharing editor. Faculty who have incorporated the CAT process into their courses have been quite satisfied and have continued to add units and/or questions to their files.

Discussion and Summary

The computer assisted testing model that has been developed has several advantages and some disadvantages for all participating. From the student's point of view, the advantages of this approach are:

(1) The self-pacing capability allows students to work through the course at whatever rate suits them.

(2) The provision of immediate feedback makes the testing activity a learning as well as an evaluation experience.

(3) The multiple testing opportunities provided for each instructional unit permit most students to achieve most of the unit objectives. Anxiety levels are typically lower than for tra[illegible]l tests; usually, the whole grade does not [illegible] one or two tests.

(4) The independent study format relieves the student of the task of listening to lectures on content he already knows or could easily learn on his own.

Some of the disadvantages, from the students' viewpoint, relate to the instructional materials used and/or the lack of alternative ways of demonstrating achievement of unit objectives. The self-pacing feature allows students to procrastinate so that there are always a few students who attempt to take all the tests during the last week of class.

From the instructor's point of view, CAT confers at least four advantages:

(1) The level of student achievement is very high. Comprehensive pre- and post measures taken over several semesters have indicated that this method is superior to the standard lecture approach used previously.

(2) The role of the instructor who uses the independent study approach changes considerably. The bulk of his time is now spent in developing and revising curricular materials (both instructional and measurement materials) and tutoring students who experience difficulty with specific concepts or techniques. The latter activity provides a great deal of useful guidance for the curriculum development phase of his work.

(3) If desired, supplementary activities which facilitate transfer of training and the enrichment of student learning can be added on. Such activities may include lectures, field trips, short term internships, research projects, etc. The point is that the CAT model, with independent study, frees the instructor of many highly routinized functions associated with more conventional approaches and allows him to engage in more innovative and profitable activities.

(4) The independent study CAT model allows the instructor to turn an entire course into an ongoing research laboratory in which he can study, in a very controlled fashion, the interrelationships which exist among CAT system variables, content variables, and student variables. The opportunity to do quality and productive instructional research is an invaluable aspect of this approach.

At present, three possible problem areas associated with CAT can be identified:

(1) System reliability is of utmost importance. The implementation of the sophisticated CAT system on a large multipurpose computer attempting to satisfy the diverse and demanding user community resulted at times in an intolerable number of system failures. Clearly, reliability is a necessary condition for the success of any CAT implementation. Fortunately, most of the problems seem to have been worked out and we feel justified in expanding our activity as additional resources become available.

(2) The security of the test item bank is of continuing concern. The CAT approach requires a higher level of security but, by its very nature, it is a very difficult system in which to maintain

security. At present, security is maintained by a maze of user and project numbers, through using the item file in packed form, and through using special keys on the CRT keyboard to permit access to the testing program.

(3) Resource allocation is a continuing problem. Once the success of such a system has been demonstrated, demand grows. The problem is further complicated by different production modes and research demands on the testing. The resolution to this problem depends heavily on the prospects for acquiring additional resources and/or reordering the priorities of the academic units involved.

As mentioned above, outstanding opportunities are afforded by the CAT process for conducting several kinds of research. A few areas for research are related to the following:

(1) Testing time is always at a premium. In CAT courses, students spend a great deal of time testing. Therefore, testing efficiency is of great concern. How can the maximum amount of information regarding student achievement be obtained in the least amount of test time? A number of researchers are currently working on this problem. Some potential answers may lie in sequential (or tailored) testing, in the application of Bayesian techniques to item selection and analysis, and possibly in the use of confidence measures.

(2) The effects of variables such as type of feedback, spacing of testing opportunities, size and nature of the item banks, setting of mastery cut points, etc., on student learning and attitudes have yet to be systematically studied.

(3) At this time, we know very little about the ways in which students use the CAT system. A catalogue of such strategies, an evaluation of their relative effectiveness, and the determination of whether or not they can be successfully used by others is clearly appropriate and important at this time.

Computer assisted testing has unquestionably expanded the repertoire of instructional strategies available to faculty. However, the full range of the effects of such strategies on student learning of and attitudes toward course content have yet to be assessed. CAT appears to have great potential; a systematic exploration of its usefulness is now in order.

BIBLIOGRAPHY

Brown, B. R., An instrument for the measurement of expressed attitude toward computer-assisted instruction. In Semi-Annual Progress Report No. R-5, Experimentation with Computer-Assisted Instruction, prepared by Harold E. Mitzel et.al., University Park; The Pennsylvania State University, August 8, 1966, 35-37.

APPLICATIONS OF DATA PROCESSING IN THE ADMISSIONS AND REGISTRAR'S OFFICE

Jack D. Thornton
J.N. Williams

Tarrant County Junior College

The organizational structure at Tarrant County Junior College provides us with the opportunity to make extensive use of Data Processing. Under our structure, the Director of Admissions and Records and the Director of Data Systems report to the same district administrative officer - the Vice-Chancellor for Research and Development.

The assistance given by Data Systems can be broken down into five primary areas: (1) admission, (2) registration, (3) add-drop, (4) student grade reporting and permanent record, and (5) institutional reports.

ADMISSION

As in all admissions offices, the application form and supporting documents may be received at different times spanning a period of several months. To accomplish the goal we set, the system had to provide the means for acknowledging the receipt of the application, provide an accessible and accurate means for answering requests about the status of an application, supply the source document for updating the application file, serve as a decision document for admission, act as a means for providing the notice of acceptance or rejection, and furnish the materials necessary for registration.

The system now in use saw its first development in the 1968 Spring semester and was implemented during the Fall of 1968. We had a major revision in 1971, and have been constantly upgrading the system since. Our admission and registration system depends on the information contained in two data bases, the student master file and the course master file. The student master file is created by merging the master file from the previous long semester with the Admission Status Report (ASR).

The Admissions Status Report is the key to our admission system and originates with the reception in the Admissions Office of the application form. The form is coded, and the data is entered by CRT's. The ASR is run daily during peak admission periods and we are able to maintain several semesters at the same time. By using the semester applied for as the control, each semester is listed separately. This is especially important during the Spring when we are receiving applications for both Summer terms, the Fall term, and possibly the next Spring term.

The admission clerks work directly from the ASR. As credentials, such as high school or college transcripts, test scores, etc., are received, their receipt is noted on the report. Each afternoon this is updated on the CRT's. The report the Admissions Office receives each morning includes not only new applicants who were added the day before, but also an update of student records that were already on the ASR. As new applicants are added to the file, a computer printed label for the letter of acknowledgment is generated and by this means we are able to notify a prospective student immediately that his application has been received.

Another report furnished the Admissions Office on a daily basis is the Admission Action Report (AAR). The credentials necessary for each method of admission are programmed, and when the required items have been received, the applicant's name is automatically placed on the AAR. The applicants folder is reviewed by an admissions clerk and the admission basis is checked, the probation code assigned, and in the case of a transfer student, the previous semester hours attempted, semester hours earned, and grade points for each college attended are coded directly on the AAR. The AAR is also updated on CRT's each day, and when this is processed the result is a computer printed letter of acceptance and an advisor card for each accepted student. When the letter of acceptance is mailed, we also include information concerning a counseling appointment and registration. When the student comes to the campus for counseling, he first goes to the Registrar's Office to obtain the advisor card and a time permit to register and then to counseling where the courses to be taken that semester are entered on the card. The advisor card and the time permit then become the student's admission to registration.

To reduce the length of the report, once accepted, an applicant's name is removed from the AAR and only one line of information is carried on the ASR.

The admission system at TCJC has proved to have several advantages. For the Fall semester of 1974, the Admissions Office was able to process almost 10,000 applications (approximately 7,000 during the month of August) with a minimum of clerical help. The reports enabled the clerical staff to answer student's questions concerning their admission status in a minimum of time.

Statistical information is entered one time, and thereafter, all letters, address tapes, etc., are furnished by the computer. An added feature is that we are able to periodically request a letter which indicates to the student the credentials necessary to complete his admissions requirements. Multiple copies of the ASR are run so that the Registrar and Counseling Offices on each campus are aware of the current status of each applicant.

REGISTRATION

Several months prior to registration, we decide on the number of time permits that we wish to have for each period on each campus. This is determined by projected enrollment and equal distribution of the flow of students for the three day registration period. The first day we register returning students only and the last two days we register new students. We do separate day registration from night registration, requesting that those who will be attending day classes, register during the day, and those who will be attending class at night, register at night.

After the student is admitted to the college, the admissions program generates a machine-printed letter informing the student that he is admitted to the college and also punches out an advisor card, which is forwarded to the Registrar's Office. As previously mentioned, the student becomes a one-line entry on the Admissions Status Report.

The advisor card has the student's name and social security number printed on it and space for schedule data to be written. The student picks up this card and a time permit to register in the Registrar's Office prior to registration and takes it to the Counseling Office where he confers with a Counselor regarding his intended course of study. The Counselor writes the department and course number on the card. This lets the student know exactly what he needs to sign up for in registration. The Counselor signs the advisor card, returns it to the student, and tells him that he must retain this card because it will be necessary for admittance to the registration area.

At the same time this is taking place, the Dean of Instruction is working with Division Chairmen and the Registrar in building the course master. The course master file and the student work file provide data for all our reports from Data Systems. The course master contains much useful information but the most unique thing is the four digit section number assigned by each Department Chairman. No other class in our entire curriculum, on either campus, has been assigned a duplicate number because blocks of section numbers are assigned to each department. Other items on the course master are: the course as listed in the catalog, the course title, building and room where the course will meet, days it will meet, semester hours credit, number of lecture hours, number of lab hours the course has, the fee code, the amount of the fee, the instructor, semester and year, campus number, the maximum enrollment limit for the course, type of funding the college receives from the state, the instructor's social security number, funding code number, the number enrolled when this particular report was run, and the corequisite requirement. (If an "L" is recorded here, this section must be accompanied by an additional course or laboratory.)

When registering, the student must present his advisor card and time permit at the admitting door. At station "1", a faculty member makes certain the student is registering at his appointed time, and if so, allows him to enter.

At station "2", returning students are given a copy of their permanent record at Tarrant County Junior College. This copy lists all courses taken, the grade earned, the grade point average, etc.

After being counseled, the student knows what courses he should take and enters the sectioning rooms, station "3", which have signs listing the departments on the outer doors. He goes to the appropriate department tables and requests what courses he wants and the desired time. The faculty member records the course, section number, days, time, building and room, and gives departmental approval. The faculty member records this information on the advisor card from a report generated by the course master called the Enrollment Status Report.

Prior to a registration period, this Enrollment Status Report is distributed to each department and taped down on sectioning tables. The Enrollment Status Report is similar to the course master, but contains information in the same order as is on our advisor card. It has the course, department, number, title, the unique section number with the suffix letter, time, days, rooms, instructor, enrollment maximum, and difference. These last three columns always allow us to know how many are enrolled, maximum allocation for the course, and how many seats in class remain. After sectioning the student, the faculty member will simply tally a one (1) to the right of the difference column by hand in order to keep up with how many have signed up for that particular course. We have a great deal of flexibility by using this report in registration to section students. It is accurate information, and we can run this report at any time.

The student goes to each department listed on his advisor card until he has completed his schedule. Afterward, he reports to station "4" for residence check. He then proceeds to station "5", which is manned by four keypunch operators who keypunch the schedule card and duplicate other cards for his packet. They will look only at the section number column on the advisor card for the unique four digit section number and keypunch these in the fields provided on the schedule card.

After the operators punch the schedule card, the student proceeds to station "6", immediately behind the keypunch station, manned by four students who simply check the keypunched card against the advisor card to verify there have not been inverted numbers or errors. The schedule card is then collected to be sent to the computer center. Approximately every five minutes, runners at station "6" carry schedule cards to the computer center.

From that point, the student proceeds to Station "7" for an I.D. All students have their pictures made and pick up I.D. cards.

The student then proceeds to station "8" to fill out auto registration card and all remaining cards, then to station "9" for health services information. At station "10", the auto registration card is exchanged for a parking sticker.

At station "11", which is directly across from the computer center, he waits for his computer schedule. In the Fall semester, when everyone has to receive a new I.D. card, chances are this schedule will be awaiting him.

When schedule cards arrive in the computer center from station "6", they are processed. The schedules are printed, separated, and sent over to station "11" where workers call out the names on each schedule until all have been distributed. No one is allowed to leave without a schedule. When the student receives his schedule, he is asked to check it against the advisor card to insure that his schedule is complete and accurate. If correct, students sign the back of the advisor card. If there is a conflict or error on the schedule printout, the student reports to station "12" for correction. After corrections are keypunched on a duplicate schedule card, the student will return to station "11" and await his corrected schedule. Once the schedule is correct, the student moves to station "13" to pay tuition and fees. His advisor card is then collected at station "14" and registration is over.

At the close of registration, the Registrar's Office receives the official registrar's card and compares data shown on it with that written on the advisor card. The registrar's card lists such information as name, address, social security number, residence status, total semester hours, audits, schedule of classes, campus, semester, date and fees.

At any time during or after registration, we can request that a Full-Time Equivalency Enrollment Report be run. This report shows the total number of students enrolled, number of semester hours they are enrolled in, and our full-time equivalence based on fifteen semester hours. Also listed will be the number of students enrolled in day classes, those enrolled in day and evening classes, those enrolled in evening only classes, and whether they are full-time or part-time.

At any time, Department Chairmen or the Dean of Instruction may add or delete classes to the course master giving us a great deal of flexibility. We might add that faculty in the sectioning rooms appreciate the face to face contact with students and the responsibility of keeping sections balanced in their own department. They generally know more about their particular departmental courses and do an excellent job advising students. Students can not section themselves because they do not know what the unique section number is until they come face to face with the sectioning faculty member. While many students could spend a great deal of time in registration, faculty and staff assisting in registration try to help them along, and discourage idleness, talking, etc., so that a large number of students may be handled in a short period of time with relative ease. During the Fall semester of 1974, we registered 8,372 students on the South Campus and 7,452 on the Northeast Campus for a total of 15,824.

Some students will spend a gread deal of time during registration in any type system; however, last Fall during our day registration when most would be full-time students, we determined that an average time of 30 minutes per student was spent. While some obviously took longer than this, others took less time to complete the registration process.

ADD-DROP AND LATE REGISTRATION

On the first class day, we begin a three day late-registration period, as well as a two day add-drop period. Students involved are handled in much the same way but in a greatly reduced space. As stated, some students will have signed up for classes "killed" due to insufficient enrollment, and others will decide they want to add or drop a course. For students registering late, the process is exactly the same as we have just described. If a student is in a "killed" class, he does not have to wait in line because the college feels obligated to assist him in getting a valid schedule.

Additionally, students are allowed to change their schedule, but must have a numbered permit representing their reserved place in line in order to be admitted to the add-drop area. This permit distributed from the Registrar's Office cuts down on long lines, soothes tempers, and allows students to report to a class or come back at a later time. It is an appreciated service to the students involved.

After entering, the student fills out a hand written form and a copy of his printed schedule is pulled. A Counselor will drop him from the course desired by entering it on the right side of the form. The Counselor gives the student the add-drop form, a copy of his schedule with a "C" on it which stands for change, and directs the student to departmental desks to add any courses he desires.

Each division contains a number of departments. Two faculty members can handle the activity for their particular division. The faculty once more have Enrollment Status Report pages for the particular courses in each department in their division, and they enroll the student by tallying him on the Enrollment Status Report as well as writing the course added in the left portion of the add-drop slip.

All of the faculty members working in add-drop are in one compact area, so we can handle around fifty students in this area at one time. If there are other changes to be made he returns the add-drop slip and the schedule to the student and directs him to the department he might be looking for. If there are no other additions or deletions to be made, the faculty member directs the student to the keypunch area.

The changes are keypunched, and forwarded to the computer center for processing. (At the same time, late registrants are being processed and this is where the card is keypunched and collected.) Again, the student goes to a waiting room where within five to eight minutes a new printout is distributed to him. During late registration and add-drop, a copy of the schedule is given to the student and a duplicate copy is retained for the Registrar's Office. Should another change take place, he will need to present his copy at the entering door. We do not want students making numerous changes, and will not allow a change without having a machine generated schedule at the entering door of the add-drop area. Should one not be available, the student is directed to the Registrar's Office where a copy is made.

Whenever students add and drop courses, instructor's class rolls will be outdated. Our program is written to generate a teacher notification card on each drop and add entered in the computer. Instructions and appropriate action is printed on the card. These are distributed daily to the faculty. After the three day add-drop period is over we do not allow students to add courses, but do allow them to drop courses until the final two weeks of the semester.

First preliminary class rolls are run the first day after registration and distributed to faculty. A second preliminary class roll is run on the fifth class day or after our add-drop period is over, so they may have a more accurate listing of students in their classes. A discrepancy sheet is attached to the second preliminary class roll. The faculty is asked to record the name of any student who should be on the roll but does not appear, and also any who are on the class roll but have never attended class. By checking this discrepancy sheet, we may find a few students who somehow are attending the wrong class. Final official class rolls are run on the thirteenth class day.

GRADE REPORTING, PERMANENT RECORD, AND INSTITUTIONAL REPORTS

To record mid-term and final grades, we distribute grade sheets (a listing of all students in each class) to the faculty. The instructor writes the grade on the grade sheet, and returns them all to the Registrar's Office. The grades are then updated on the CRT's by the secretaries in the Registrar's Office. Four copies of each current student's permanent record card are printed. The first or original copy is microfilmed and placed in student's fiche jacket and used to run transcript requests. The other copies are marked with a phantom design, "for student use only", and are distributed to the Registrar's Office, Counseling Office, and to the student who receives it as his semester grade report. Transcripts can be mailed on request the next working day after grades have been turned in.

Additional reports which are run from the student master file and the course master file are: student course load, concurrent enrollees on each campus, audit students, building and room utilization, Coordinating Board reports 001 and 003, Full-Time Equivalency, add-drop analysis, grade analysis, probation-suspension reports, students by major, honor reports, error statistics during registration, a statistical analysis, excess hours report, Student Profile Report, Vocational-Technical Report, and Fee Audit Reports.

In summary, the major advantages of the system are as follows:

1. Keypunch input compares favorable to the pull card method when considering accuracy.
2. Flexibility in updating the course master file.
3. Fast turn-around time in processing schedules, class rolls and reports.
4. Registration is confined to a three-day period.
5. Students receive a machine printed copy of their schedule.
6. Students spend very little time in the complete registration process.
7. One-way flow in only one building for the complete process.
8. Every student has an accurate schedule with no conflicts or omissions.
9. System is adaptable and compatable with the add-drop process.
10. The system is simple to implement and control.
11. All fees have been accurately collected.
12. Only one card is used to generate the student schedule and none are used to generate class rolls.
13. All of the faculty are not required to assist in registration for the full three-day period.
14. Our add-drop transactions represent less than 10% of our total head count.
15. We can stay with this system through our expected large enrollment increases in the future.

All colleges have their own admissions, registration, and reporting systems and all do what they feel is best for them. We don't propose that any college change to what we are doing. We are convinced, however, that in our particular situation, the system discussed in this paper does what we want done the easiest, quickest, most accurate way. We will continue to make improvements and refinements as needed and as circumstances allow.

Glossary of Terms

ASR (Admission Status Report) - listing of all applicants for a specific semester which also includes items necessary to complete their file

AAR (Admission Action Report) - listing of all applicants for a specific semester who have turned in all items required for admission

CRT (Cathode Ray Tube) - IBM Video Terminal Display

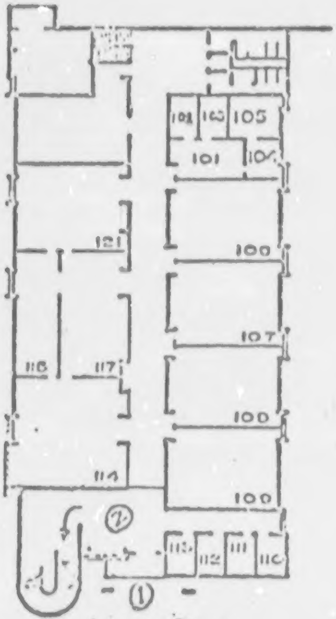
Course Master - listing of all courses offered for a specific semester on a specific campus. This file is essential in our control of registration and student records.

ESR (Enrollment Status Report) - report generated from the course master used in registration as a control of classes. A new printout is run after each registration period.

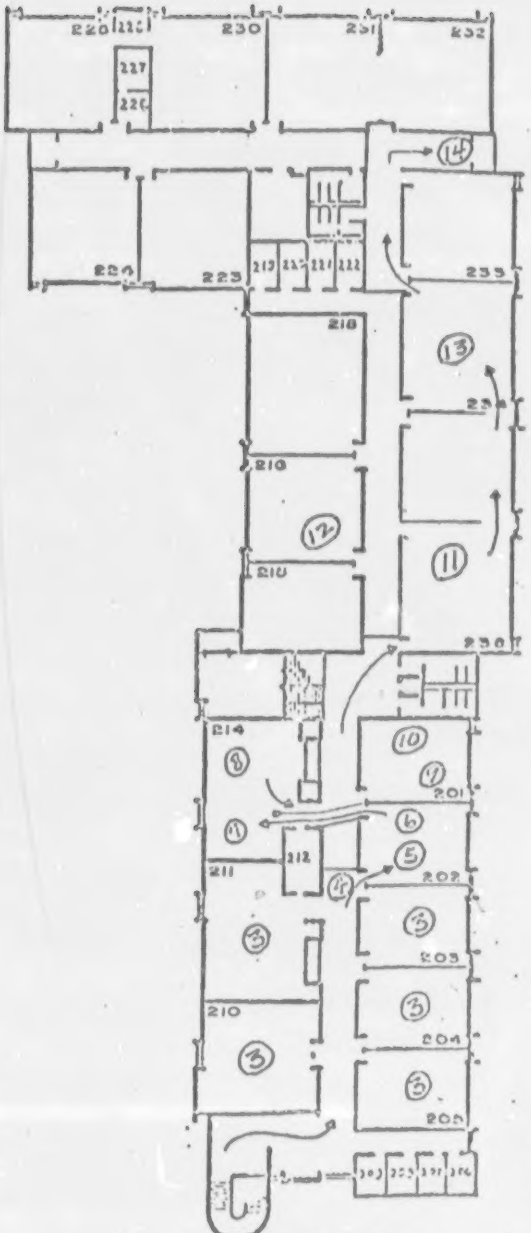
FTE (Full-Time Equivalency Enrollment Report) - report that shows accurate data of registration, number of students enrolled, full-time, part-time, day, night, etc.

Coordinating Board Reports - required by the State of Texas for funding and statistical information

Registration Flow Chart


102
103
105
101
104
100
121
107
100
118
117
114
100
②
113
112
111
110
①

Lower Floor


228
229
230
231
232
227
226
⑭
224
223
219
220
221
222
235
218
⑬
234
216
⑫
⑪
215
236
214
⑧
⑩
⑨
201
⑦
⑥
212
⑤
④
211
202
③
③
203
210
③
③
204
③
205

Upper Floor

CAA: COMPUTER ASSISTED ATHLETICS

John H. Hall, Assistant Director
Computer Research Center
University of South Florida

The proliferation of computer and computing technology in the past ten years has had an effect on nearly all academic disciplines. Certainly, athletics and related programs are no exception. The utilization of computing services as well as the number of applications supporting athletics is low when compared to more traditional computer related activities (e.g., engineering, science, etc.); however, very much on the increase.

The indoctrination of students in high school and college to computers and their technology has resulted in orienting people of all disciplines to the merits of computing. This, coupled with the increasing availability of hardware, has provided the opportunity to apply the valuable resources of automation to support athletics. National exposure of professional athletics and the frequent references in the media to the pro's use of computers have also served as catalysts to many people responsible for athletic programs to investigate the computer's potential for their situation.

The use of computers by athletics and their related activities is a very recent phenomenon. However, attempts were made as early as the late 1950s to integrate computing into sports. A baseball team tried replacing the manager concept of running their team with rotating head coaches. They also used the computer to solve such problems as which relief pitcher to call in a certain situation, and probably which candidate for head coach should be next in line when the incumbent failed to yield a winner. They had several problems among which were: (a) the lack of good ball players, (b) the attitude that players should play the game without the interference of computers, and (c) the lack of credibility concerning the information furnished by the computer.

Several other experiments were tried through the early and mid 1960s. Most were indeed exploratory in nature, and required a high level of computer sophistication on the part of the user such that few, if any, were announced as successful. The introduction and ultimate availability of third generation computers, their terminal capabilities, and more usable high level languages brought a whole new base from which to apply computing to athletics and sports. Today, such things as conversational terminals, special purpose mini-computers, availability of data base technology, the potential of medium-large scale central processing units, and breadth of knowledge in the application of computing have served to provide sports and athletics with some many unique and creative systems.

Certainly one of the more common uses for the computer in support of sports and athletics is the processing of administrative data. Accounting, budgeting, and other business type functions are assisted by data processing systems both for professional and amateur programs. Such things as computerized season's ticket files and lists are often used when the number of tickets involved is very large. Most of the systems in operation that perform such functions as listed above are indirectly supportive of the athletic programs. The main theme of this paper will however, deal with applications of more direct support for athletic contests and athletes through the use of computing.

Research efforts assisted by computer modeling and testing have had a very significant impact on various sports programs. A recent television special dealing with football injuries devoted a large portion of time to the portrayal of study efforts aimed at improving the design of the football helment. Several major universities were involved and explained their research techniques which involved a great deal of modeling and impact study analysis via computer. Should these studies result in a safer helmet, as it appears they will, the impact on the football player will be very direct and beneficial.

Another example of beneficial research that could lead to improved performance was done by Professor James B. Vernon, associate professor of mechanical engineering at the University of Southern California. Professor Vernon has designed and built a pole-vaulting pole with a bend in it.[1] He used the computer to help solve the very complex energy problems involving the motion of the vaulter, energy in the pole, etc. Optimizing on his theory and using a proper bent pole design and vaulting technique, Professor Vernon predicts the theoretical possibility of a 28½ foot vault.

Contests have been directly affected by the use of computers in many ways. Conferences have scheduled their officials via computer, and used the same systems to evaluate their performance. Several specific examples of computer driven scoring and display devices have been publicized recently. A couple of years ago an IBM 1130 was mounted in a van and made the professional golf tour. The side of the van served as the scoreboard and the computer took care of keeping the scoreboard up to date as the scores for each hole came in. Certainly one of the most modern score/message boards in sports today is at the Harry S. Truman complex in Kansas City. Joe Garagiola of NBC did one of his specials on the operation of the baseball score and message board and anyone watching had to be impressed.

---

[1] James B. Vernon, "Pre-Bent Pole Looks Promising," The Bent of Tau Beta Pi, Fall, 1974, pp. 14-15.

The professionals are not the only ones to use computers to assist in contests regulation and scoring. Lakeville Junior High, Lakeville, Minnesota used its interactive computer on the TIES (Total Information Education Systems) network for such a purpose.[2] They set pairing by weight, grade, and school for a wrestling tournament of nine teams and 150 wrestlers. Not only did the automation of the pairing process save time, it was far more equitable in its scheduling and matching of the wrestlers. Also, the 32nd Annual NCAA National Gymnastics Championship held at Pennsylvania State University used an interactive terminal system to tabulate the scores for its 43 participating institutions, 180 performers, and 5,000 individual scores.[3] The intricacies of the gymnastics scoring processes were all coded into the main computer, so team and individual scores and totals were brought up to date immediately upon entry. This allowed officials, coaches, and others to gain current status reports upon demand throughout the session.

Professional football is probably one of the most prolific users of computer systems in sports today. Nearly all of the pro teams use computerized scouting reports to assist them in selecting players in the draft.[4] Player scouting is by no means the limit of pro-football's computer assistance. Opponent's tendencies on offense and defense as well as studying one's own tendencies are among the many applied users of the computer in game planning.[5] A more thorough explanation of a game scouting program and its uses will be covered later in this paper.

Many other examples could be listed from nearly all competitive sports in which the computer is used in one form or another. Specifics on many of these applications are kept very confidential by the teams using them. The advantage of a team knowing the specific breakdown of scouting data used against them would undoubtedly be a valuable asset. Having been closely involved with the development of two systems used directly to support athletics, it's very clear that such systems can be very valuable to a sports program. The description of these two systems follow:

Athletic Injury Report System

The use of the computer to break down data on athletic injuries is a natural. The process is a reasonably mundane task and very time consuming if attempted manually. However, the computer can and will perform this analysis with extreme accuracy and speed. The trainer and his staff no longer will need to spend many hours of manual processing to produce reports of their activities. These manual reports, at best, just scratched the surface of providing useful information. The use of the computer in the analysis process will allow a greater number of data items to be considered and yet require less personnel time to break down.

The procedure used at Northern Illinois University the past several years and adopted in modified form for the 1973 year by seven universities in the Mid-American Conference is based on the collection of injury data through the use of an optical mark sense (OMR) readable sheet.[6] (Figure 1) This sheet is used in the training room and data is collected for each athlete requiring the trainer's attention. Specific data items are collected about the injury and its treatments. There are two items on each side of the sheet that are somewhat unrealted to the injury itself, but absolutely required to machine process the sheets. These two items are the athlete's social security number and the injury number which is merely a sequence number denoting how many injuries the player has had requiring visits to the training room. Each of these elements is required on both sides of the sheet to insure the mechanical process of matching the information on the two sides of the collection sheet into a single machine processable record.

From looking at the sheet, it is obvious that the process of recording the data is not an insignificant task. And, indeed, a good bit of the time formerly spent on reducing injury data manually can now be used in recording the data on the OMR sheet. There are probably several justifications and rationalizations for reinvesting this time. First of all, the sheet has the capability of recording a significantly larger number of data items than are usually recorded under a manual system. The more data collected, obviously the more in depth analysis available on injuries and their treatments. The time spent coding the OMR sheets can be spread through the entire season; whereas, the manual compilation of an injury report at the end of the season would concentrate a great deal of time within a few days or weeks. Also, after using the OMR sheets for several weeks the process of recording on them becomes increasingly faster and easier.

Analysis of the injury data requires several steps. These steps are:

(1) Processing of completed forms through the OMR reading device and converting data to computer processable form.

(2) The execution of several computer programs which match the two sides of the injury form's data produced in the previous step

---

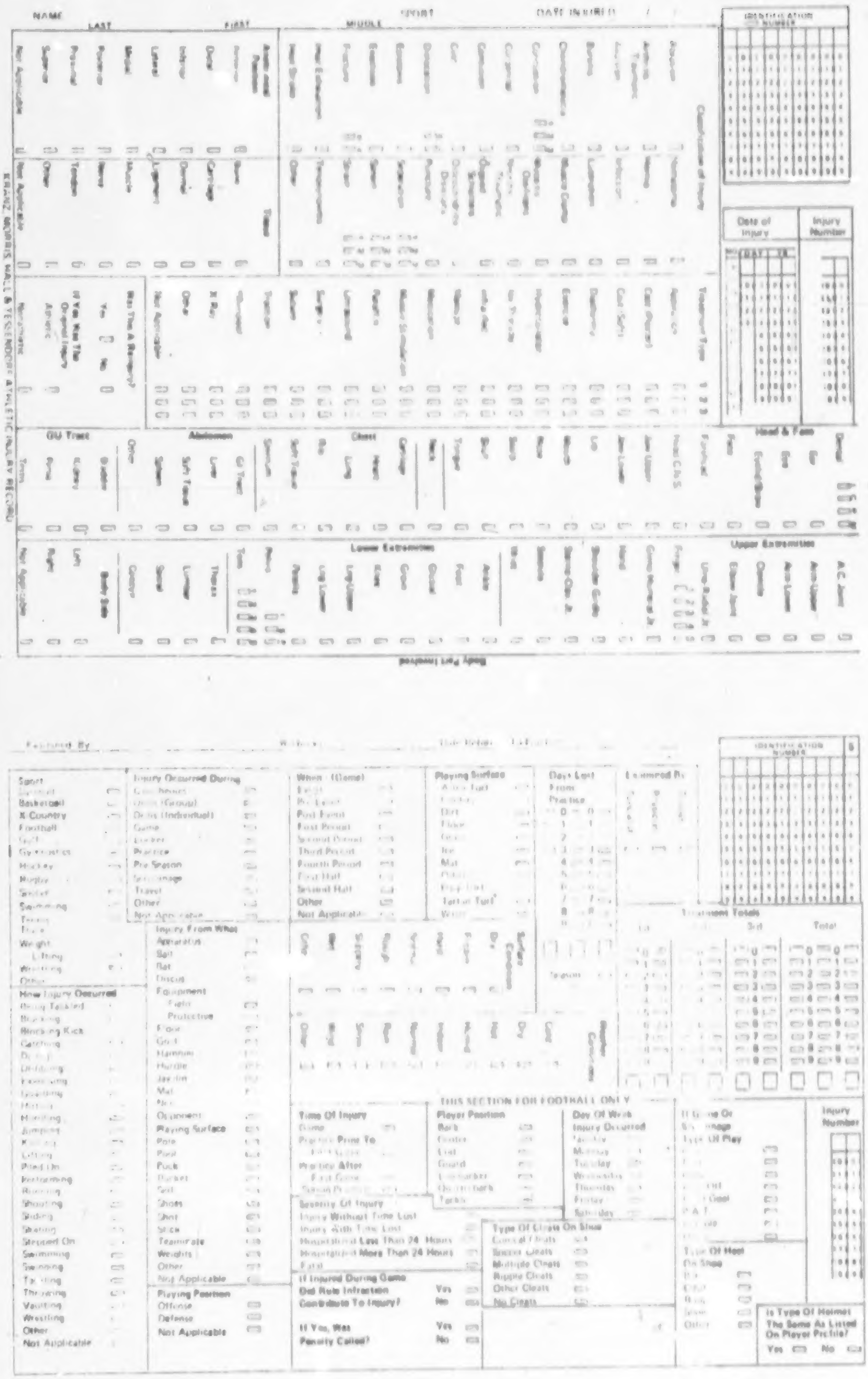
[2] Richard Rowe, "Matching Up the Grunt and Groaners," AEDS Monitor, April, 1974 page 4.

[3] "Meet's Terminal Scores Athlet's Feats," Computerworld, December 4, 1974, page 8.

[4] Ward, Gene, "Ward to the Wise," New York Daily News, October 21, 1971, Sports Section.

[5] Kyle Rote, "Computer in the Starting Lineup," Think, October, 1973, pp. 24-28.

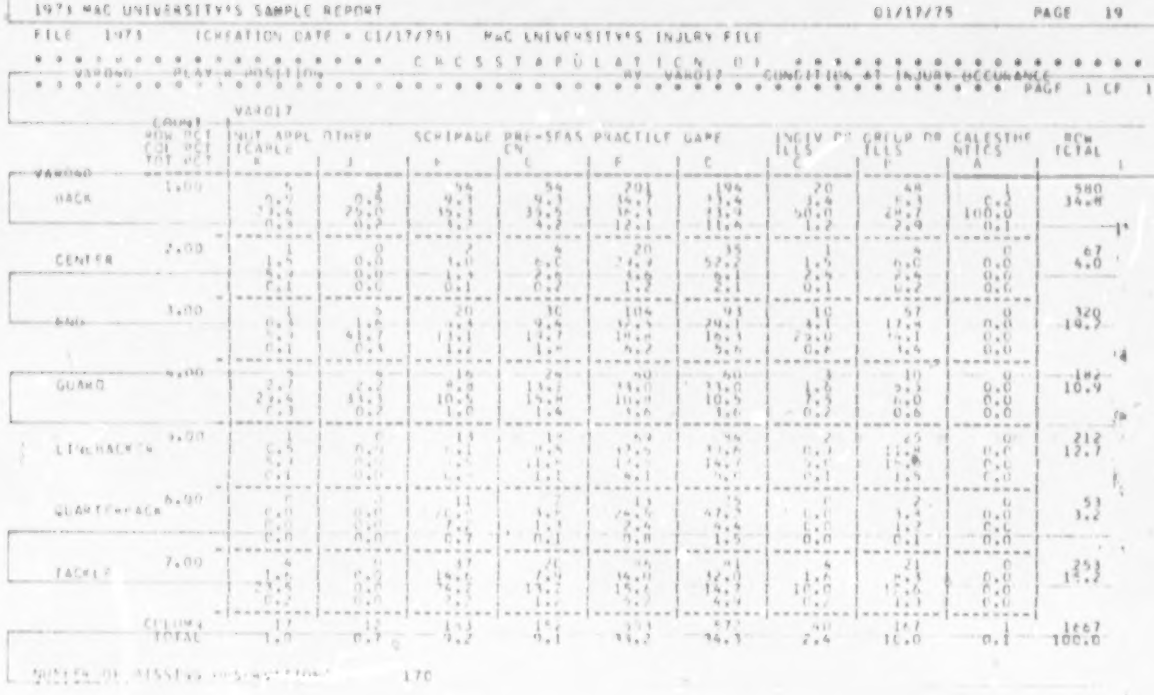
[6] John Hall, Alfred Kranz, Harold Morris, William Tessendorf, J. Randall Ryan, "Northern Illinois University Athletic Injury Report System," Athletic Training, December, 1974, pp. 184-185.



Side 1

Side 2

Figure 1



```
1975 MAC UNIVERSITY'S SAMPLE REPORT                                   01/17/75        PAGE   19
FILE   1975      (CREATION DATE = 01/17/75)    MAC UNIVERSITY'S INJURY FILE
* * * * * * * * * * * * * * * * *   C R O S S T A B U L A T I O N   O F   * * * * * * * * * * * * * * * * *
      VAR040     PLAYER POSITION                          BY  VAR017    CONDITION AT INJURY OCCURANCE
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   PAGE  1 OF  1

NUMBER OF MISSING OBSERVATIONS =        170
```

Figure 2
Sample Injury Report

and the creation of a single computerized record for each injury sheet.

(3) One or more computer runs to analyze the records produced through a generalized statistical package.

Specifically, N.I.U. uses a Digitek 100 OMR mark sense reader with a magnetic tape unit to record the data in machine readable form. A utility program sort is used to order the records for each side of the sheet in social security and injury number order. A FORTRAN program is then run to combine the records for each side of the sheet and will produce an error report for sheets which do not have matching sides.

The Statistical Package for the Social Sciences (SPSS) is then used to break the data into usable information. SPSS is very convenient to use in that very little computer expertise is involved in its use. With 2-3 hours of training most anyone can learn enough about SPSS to set up the control cards for the analysis required. The ability of SPSS to combine data elements in cross-tabulating is just one of the uses for breaking down the injury data. An example report available might be a cross-tabulation of football player's position by conditions at the concurrence of the injury (Figure 2)

It will now be possible although the programs are not yet written, to save a complete injury file for each competition year. Then an individual injury profile can be produced for each athlete at the end of his eligibility showing each injury and treatment from his first practice through his last game. This will be of immeasurable value in assisting the trainer to advise professional scouts about a player, and in planning for prevention of injuries.

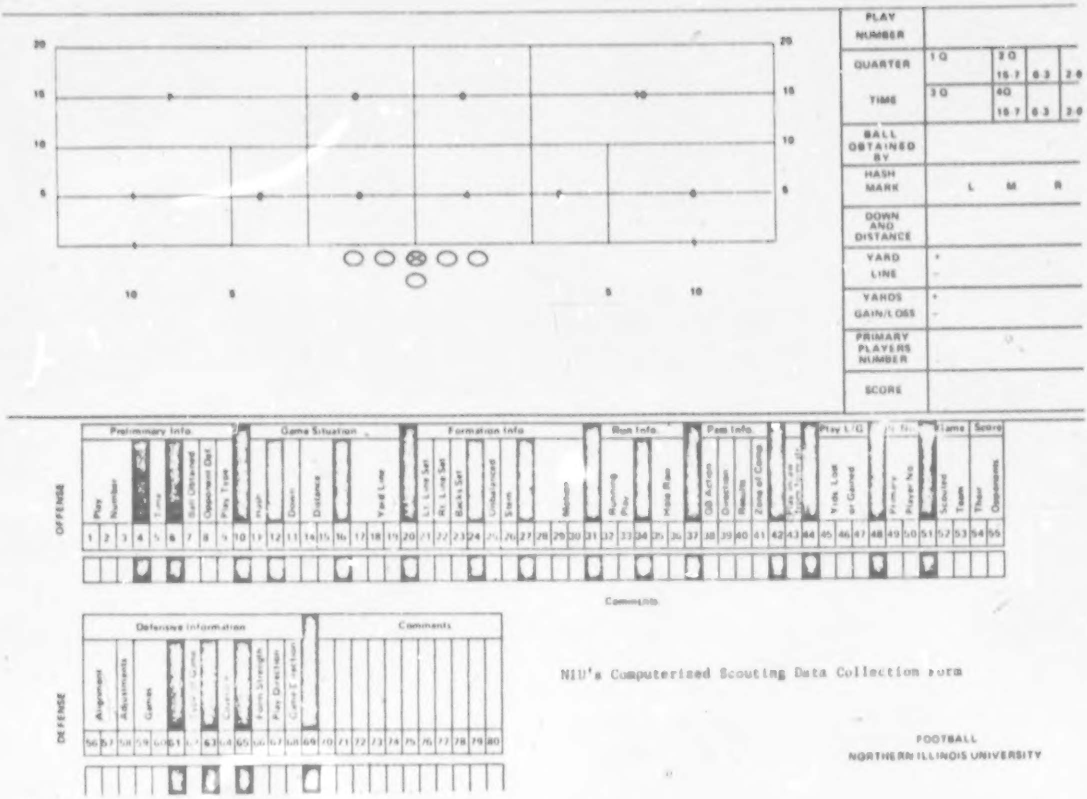
In addition, it is entirely conceivable that the availability of this data can lead to some very significant research into the study and prevention of athletic injuries. Should the collection and synthesis of this data help in reducing the quantity and severity of athletic injuries any at all, it will have been worth the investment of time and energy involved in its implementation and operation.

Football Scouting System

In the fall of 1969 the football coaching staff at Northern Illinois University (NIU) used an edged punched card system for scouting their opponents. After having spent some 40-100 man-hours per week to break down scouting data, they decided there had to be a better way. The coach responsible for coordination of scouting came to the NIU computer center and asked for help. In studying the problem it was discovered that computerized scouting systems existed at other institutions. However, the decision was made to design a new system since those in existence were either unusable or unavailable.[7]

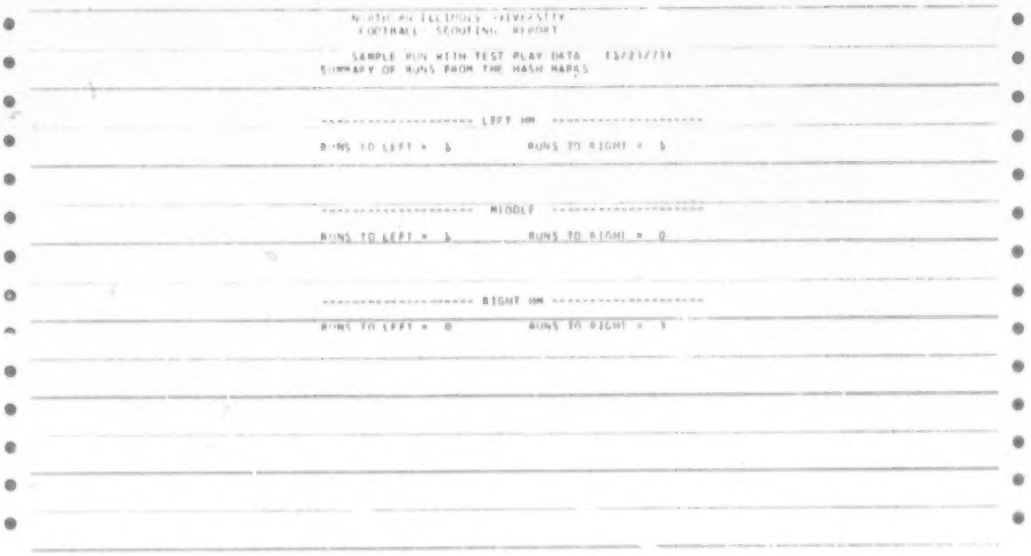
---

[7] J. H. Hall, "A Descriptive Analysis of a Computerized Football Scouting System" (unpublished Master's thesis, Northern Illinois University, 1972)


PLAY NUMBER
QUARTER
1Q
2Q
TIME
3Q
4Q
15-7
6-3
2-0
BALL OBTAINED BY
HASH MARK
L
M
R
DOWN AND DISTANCE
YARD LINE
YARDS GAIN/LOSS
PRIMARY PLAYERS NUMBER
SCORE
OFFENSE
Preliminary Info.
Game Situation
Formation Info
Run Info.
Pass Info.
Play L/G
Game
Score
Play Number
Time
Ball Obtained
Opponent Def.
Play Type
Hash
Down
Distance
Yard Line
Lt. Line Set
Rt. Line Set
Backs Set
Unbalanced
Slant
Motion
Running Play
Hole Run
QB Action
Direction
Results
Zone of Comp.
Yds. Lost or Gained
Primary Player No.
Scouted Team
Their Opponents
Comments
DEFENSE
Defensive Information
Alignment
Adjustments
Games
Form Strength
Play Direction
Game Direction
Comments
FOOTBALL
NORTHERN ILLINOIS UNIVERSITY

NIU's Computerized Scouting Data Collection Form

Figure 3



```
           NORTHERN ILLINOIS UNIVERSITY
            FOOTBALL  SCOUTING  REPORT

           SAMPLE RUN WITH TEST PLAY DATA   (5/21/79)
      SUMMARY OF RUNS FROM THE HASH MARKS


    -------------------- LEFT HM --------------------

    RUNS TO LEFT =  1           RUNS TO RIGHT =  1


    --------------------  MIDDLE  -------------------

    RUNS TO LEFT =  1           RUNS TO RIGHT =  0


    -------------------- RIGHT HM -------------------

    RUNS TO LEFT =  0           RUNS TO RIGHT =  1
```

Figure 4

The design of the scouting system was based on a thorough set of specifications for the data to be collected and reports required. The list of data elements to be used was finalized and a general purpose scouting data collection form created. (Figure 3) This form serves as both the data collection device for the scout and as an input document to a keypunch operator.

Programming of the system was done in the IBM FORTRAN IV G. The program contains three distinct sections. These sections are: for storage allocation and definition; the reading, editing, and storing of data; and report generation.

Storage allocation and definition initializes the descriptor information and allocates data storage and work space in the program. The next section reads the data cards, edits the data, and stores the valid data in the space allocated by the previous part of the program. The reports are then generated and printed. The reports generated are as follows:

1. Chronological List of Offensive Plays
2. Summary of Running Plays by Play Type
3. Summary of Running Plays Into and Away from Strength.
4. Summary of Pass Plays Into and Away from Strength
5. Summary of Plays by Backfield Alignment
6. Summary of Tendencies from the Hash Marks (Figure 4)
7. Summary of Plays by Line and Backfield Formation Combination
8. Summary of Plays by Down and Distance
9. Summary of Running Plays Through Each Hole
10. Summary of Pass Plays to Each Receiving Zone.
11. Summary of Each Principal Player's Play
12. Summary of Offensive Plays by Field Position
13. Summary of Backfield in Motion Plays
14. Chronological List of Defensive Plays
15. Summary of Defensive Success Against Running Plays Through Each Hole
16. Summary of Defensive Success Against Pass Plays to Each Receiving Zone
17. Summary of Defensive Alignments by Field Position

The computerized football scouting system requires human judgment and interpretation in analyzing the reports produced. The operation of the system provides for scouting data to be reduced into a much finer breakdown than the manual method within the time constrain's involved. The coaching staff found their time investment in breaking down computerized scouting reports to be between 10% and 40% of what it was with the previously used edged punched cards. The computer runs take about 10 cpu seconds per game on an IBM 360/67. The NIU system would probably require minor modifications to match the naming conventions of another coaching staff, but would be usable to another team with similar report requirements.

In summary, it is certainly clear that computers are now an integral part of athletics. There are certainly more applications than have been referenced in this paper. Just as certainly, the creativity of computer people, coaches, administrators, and athletes will lead to many more useful applications for the computer in the future to help play the game.

BIBLIOGRAPHY

Hall, John H. "A Descriptive Analysis of a Computerized Football Scouting System." Unpublished Master's thesis, Northern Illinois University, 1972.

Hall, John H., Alfred Kranz, Harold Morris, William Tessendorf, J. Randell Ryan "Northern Illinois University Athletic Injury Reporting System." Athletic Training, December, 1974, pp. 184-185.

Meet's Terminal Scores Athlete's Feats" Computerworld, December 4, 1974, p. 8.

Nie, Norman, Dale H. Bent and C. Hadlai Hull, Statistical Package for the Social Sciences, New York: McGraw-Hill Book Company, 1970.

Rote, Kyle, "Computer in the Starting Lineup" Think, October, 1973, pp. 24-28.

Rowe, Richard, "Matching Up the Grunt and Groaners" AEDS Monitor, April, 1974, p. 4.

Vernon, James B., "Pre-Bent Pole Looks Promising" The Bent of Tau Beta Pi, Fall, 1974, pp. 14-15.

Ward, Gene, "Ward to the Wise" New York Daily News, October 21, 1971, Sports Section.

IMPROVEMENT OF SCHOOL DESIGN THROUGH THE SIMULATION OF EDUCATIONAL ACTIVITIES

William S. Bregar
The Pennsylvania State University

1. Introduction

This paper will describe research into the nature of the design of educational facilities and the development of a computer simulation system to help solve problems in designing educational floor plan layouts. Such a system can be used by school designers to evaluate and improve school floor plans.

The implementation of the system involved 1) the design and implementation of a procedure for determining those characteristics of a school program which affect or are affected by the physical constraints of a school building, 2) incorporation of those characteristics into a model which could be applied by a simulation program to a proposed floor plan, and 3) the development of computer programs which could successfully simulate educational activities on the proposed floor plan and provide feedback to the designer as to the sufficiency of its design. The focus of this study was on elementary school facilities.

2. Design Problems in Education

"The design of school must be consistent with the type of instruction systems carried on" (Engelhardt, 1970). The evolution of teaching methodology has historically been accompanied by changing school structures to accommodate them. The one-room schoolhouse of the nineteenth century has been transformed to the egg-crate construction of the first half of the twentieth century and finally to the open-plan schoolhouses of the past decade. The egg-crate design came about as a response to the requirement of absolute independence between grade levels; the open-plan in response to the attempt to abolish grade level differences and open lines of communication among teachers and students (Leu, 1965).

Although general changes in architecture have come about with educational innovation, the relationship between the detailed aspects of an educational program and the instructional space has not always kept pace with such changes.

In fact, it is not clear that the relationship between emerging elementary educational methodologies and the use of elementary school space is very well known. An examination of award winning schools in 1967 (Nation's Schools, 1968) showed a range of from 54.2 to 112.3 square feet per student with costs ranging from $10.80 to $33.84 per square foot. Instructional space ranged from 40% to 80% of the total facility. The majority of these schools were of an open plan design however, the educational program to have been implemented was not specified.

Some attempts to analyze the prospective use of school space have been made in recent years. Engelhardt (1970), for example, considered the activities which would likely go on in a continuous progress open plan school, the factors to be considered in planning such a school, and the relationship between spaces, then prescribes space requirements to accommodate the program. Square feet for each kind of space is presented, however, the method of determining these requirements is not described.

Banghart and others at Florida State University (Banghart, et al., 1970) describe a model for defining space needs in high schools utilizing a building block approach. Each basic unit (student module) is defined as the "space and resources required to maintain a student in a given activity at a particular time." Space needs are determined by simulating schedules of activities based on previously compiled information on student requests for activities, duration of the activities, and the number of activities requested [illegible] student.

Apker (1970) using available seating as his spatial denominator simulated a high school with modular scheduling, given an architect's plan for a proposed high school. Utilizing the Generalized Academic Scheduling Program, (Murphy, 1970), Apker showed that for projected enrollment of 1500 students, the number of classrooms was 21 more than were needed and that seating space for large group instruction was "under-estimated."

The Florida State University and Apker projects do provide some basis for analyzing floor plans as they apply to high schools. However, the problem of varying methodologies does not appear as a factor in their analysis of the use of space. That is, by assuming the traditional mode of education, space needs can be translated from seating requirements. These projects also had the advantage of being able to work with fixed regular schedules which would remain in effect for an academic term. Thus, while sufficient for high school environments, such techniques might not be adequate to analyze space needs under circumstances where schedules could change from day-to-day and where methodologies could affect the sizes and kinds of student groupings.

The research described herein took place in two stages. First, a method was derived to ascertain whether there were indeed differences in the use of instructional space by differing educational programs. Second, given a verification of the first premise, a computer system was developed and provide summary information of their impact on a proposed floor plan design.

Although the emphasis in this paper is on the computer system, it should be mentioned that the first stage consisted of an observation of three elementary schools which operated under two different educational systems. Two could be characterized and a traditionally oriented schools[1], the other was organized as a multiunit[2] school. Two main results were obtained from the observation data. One was that for the schools observed, group size varied significantly as a function of school program and activity; the other was that activity duration did not appear to vary significantly for these factors (Bregar, 1974). This information would be utilized in the generation of schedules of activities for an elementary school program.

The data collected during the observation phase was reduced to provide information to the simulation program about schedules and characteristics of activities for a given methodology (in this case traditional or multiunit).

3. System Design

A system of computer programs was written which could simulate activities on a proposed school floor plan. The model employed was stochastic and discrete and was based on the data collected during the observation phase. The overall system design for the simulation is shown in Figure 1. Floor plan specifications are digitized and processed into a hierarchical structure representing the space-subspace relationship found in observed instructional space. Included in the specifications are the inventories of furniture and equipment for each space.

Observed data are compiled into frequency tables called *activity descriptors*, which are used to generate certain physical characteristics of activities. The activities themselves are generated from a *block schedule* which is input to the simulator. The *block schedule* represents a school day which is typically divided into four or five blocks of time, up to two hours each, during which a major subject area is pursued. An example of the block schedule is shown in Table 1.

To complete the information necessary for a simulation run, parameters must be input which delineate the school environment to be simulated. Specifically, the parameters are 1) school program type,[3] 2) mod length, or the length of the scheduling module, 3) proposed enrollment, and 4) a designator declaring whether optimal or observed space figures should be used in computing space use.[4]

The functional flowchart (Figure 2) portrays the simulation system. A preprocessing stage converts the inputs to an internal format consisting mainly of a hierarchical floor plan representation and a set of Activity Control Blocks (ACB's). The ACB's are created by a schedule generator which generates a set of subactivities for each activity in the block schedule. The ACB (Figure 3) contains all pertinent information about a subactivity such as its scheduled time and duration, the number of students assigned to it, furniture and equipment requirements, and total space requirements. When an activity is assigned to a space, its ACB is updated to reflect its location. Essentially, the schedule of subactivities allocates time from the block schedule and students from the available enrollment in such a way as to reflect grouping practices employed in the school program being simulated.

The simulation process itself uses a time-slice approach whereby time is initialized at the earliest time on the schedule and incremented by regular amounts (the mod lengths) until the end of the schedule is reached. At each interval, terminating activities return their resources and students to their respective eligibility pools. New activities are assigned particular students from the available student pool and characteristics are generated reflecting the information contained in the file of activity descriptors. These attributes are generated reflecting the information contained in the file of activity descriptors. These characteristics include: 1) furniture and equipment required, 2) space per student, 3) a distraction[5] factor, and 4) a group type indicator specifying whether a group is working as a group or as individuals and whether the group is supervised or unsupervised.

When the new activities have been completely characterized, the program makes an assignment of activities to spaces in such a way as to find a reasonable fit between the activities and the spaces which are available for them. The problem of assigning activities to spaces is a particular instance of the linear programming assignment problem which is not solvable by normal methods because of the space-subspace relationship on the input floor plan. An algorithm which finds approximates to optimal solutions can be found in Bregar (1974).

Once activities are assigned to spaces, the ACB's of the new activities and those in process at the current time interval are scanned and space utilization figures are compiled. At the end of the simulated day, all of the space utilization figures plus other compiled information is output for analysis. The school designer can then modify his design, if appropriate, and restart the simulation.

Outputs from the program include the generated schedule of subactivities and their generated characteristics including furniture and equipment requirements. In addition, at each time interval a summary table is printed which gives for each activity starting or in process:

1) the activity name
2) start and end time of the activity
3) number of students assigned to the activity (an optional listing of students by student number is also available)
4) area of space required by the activity
5) name of the space assigned to the activity
6) area of the space assigned
7) ratio of space utilized to space available for the activity.

Following the list of scheduled activities, the percentage of space in use of the total available instructional space is computed for each time step. At the end of the day, the percentage of use of each space in terms of its total time available is output.

---

[1] Although there has never been a single monolithic system which could be defined as a traditional methodology, there are certain characteristics which could be generally attributed to the concept. Among these are 1) a one teacher per class organization, 2) an age graded division of students, and 3) an emphasis on teacher-centered group instruction.

[2] The multiunit school concept is based on 1) a team of teachers administering one of several independent units in a school, 2) a non-graded division of students, and 3) individual, one-to-one (student/teacher) small group, and large group instruction emphasizing the student as an individual. (Klausmeier).

[3] The program will currently accept multiunit and traditional program designators.

[4] Space use can be computed either by employing figures from the observation phase or by using a selection of derived optimal square feet per student estimates based on furniture and configuration combinations.

[5] The distraction factor is an indicator of the potential for an activity to distract an activity in an adjacent space with no intervening walls.

With these ouptuts a school designer can evaluate the way the spaces he has designed will likely be used. If he wishes, he can then adjust the floor plan, or any of the parameters to see their effects before deciding upon a final design.

4. Results

An example of how the program can be used is presented in this section. Figure 4 represents the floor plan of a pod from an open plan school which was observed during the observation phase of this project. The dotted and dashed lines depict observed subdivisions of the space in which activities were carried on. The space was designed for a maximum enrollment of 150 students under the multiunit program. The hierarchical relationship of spaces is evident in the figure. The area of the entire pod is 7025 sq. ft. and subspaces range from 41 to 1815 sq. ft.

Assuming the floor plan configuration shown, the question could be asked, "How would this floor plan function if its enrollment were 200 instead of 150 students?"

To establish a benchmark, the simulation program was run so as to simulate the observed school with its observed enrollment on its own floor plan. The complete output which is given in (Bregar, 1974) showed that the generated schedule of subactivities and their characteristics was representative of what had been observed. Increasing the enrollment by 1/3 resulted in an expected increase in the number of subactivities for each activity. There was a consequent general increase in both the number of spaces which were put into use as well as the number of modules for which a given space was occupied. This is evident from the space usage summaries shown in Table 2 and Table 3. However, many of the spaces were used less than 60% of time available -- some of these, of course, were not available when their parent spaces or subspaces were occupied. Out of 117 scheduled subactivities only 13 were not assigned spaces. It should be noted that there were spaces available at these times, however, they were not considered suitable by the assignment algorithm. With this in mind, the conclusion can be drawn that there is adequate space available for 200 students in the pod.

A second test run was made utilizing a 150 student enrollment but removing space C27 from a floor plan. (C27 has been shown on the benchmark run to be a lightly used space.) Results from second run indicated that the activities assigned to C27 could have been assigned to other available space with no other noticeable difference in the functioning of the school.

Space C27 had an area of 165 sq. ft. At the cost of $20 per square foot, elimination of this space could have resulted in a savings of $3300 in the cost of the school.

5. Summary and Implications

A description of a system for simulating the activities of an elementary school operating under a particular educational program has been given. Through direct observation, a model for an educational program was prepared and computer programs were described which could apply a specified model to a proposed school floor plan. Results from the program can be used to evaluate the functionality of a floor plan, which, if necessary, can be revised and reevaluated before building a school which might not meet the needs of the proposed educational program to be implemented.

REFERENCES

Apker, Wesley L. *The use of simulation in determining the effects of certain program changes on room utilization.* Ph.D. Thesis, University of Washington, May 27, 1970.

Banghart, Frank W., et al. *Simulation for determining student station requirements and school space needs.* Technical Report. Tallahassee, Florida: Florida State University, Educational Systems and Planning Center, Document G30701, 1970.

Bregar, William S. *Computer-aided-design of educational facilities.* Ph.D. Thesis, University of Wisconsin, September, 1974.

Engelhardt, Nikolaus L. *Complete guide for planning new schools.* West Nyack, New York: Parker Publishing Company, inc., 1970.

"How 1967 award winning schools compare." *Nation's Schools,* 81, 1 (January, 1968), pp. 41 - 70.

Klausmeier, Herbert J., Mary Quilling, et al. *Individually Guided Education and the Multiunit Elementary School* Experimental Copy. Madison, Wisconsin: Wisconsin Research and Development Center for Cognitive Learning, University of Wisconsin.

Leu, Donald J. *Planning educational facilities.* New York: The Center for Applied Research in Education Inc., 1965.

Murphy, Judith. *School scheduling by computer.* New York: Educational Facilities Laboratories, 1964.

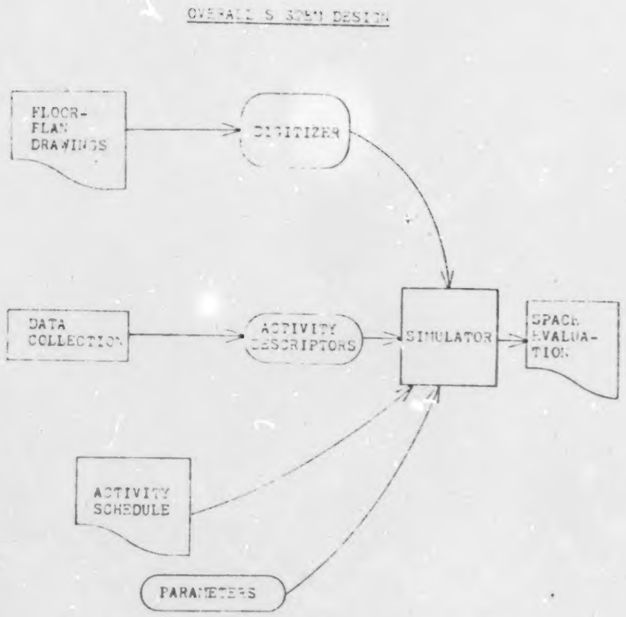

OVERALL SYSTEM DESIGN
FLOOR-PLAN DRAWINGS
DIGITIZER
DATA COLLECTION
ACTIVITY DESCRIPTORS
SIMULATOR
SPACE EVALUATION
ACTIVITY SCHEDULE
PARAMETERS

Figure 1.

Table 1

Sample Block Schedule

8:30	Language Arts Block		
		9:00- 9:30	Physical Education-Group 1*
		9:30-10:00	Physical Education-Group 2
10:30	Math Block		
11:30	Lunch Hour		
12:30	Study Hall		
		12:30-1:00	Music Group
		12:30-1:00	Physical Education Group
		12:30-1:00	Art Group
1:00	Math Block		
2:00	Recess		
2:15	Language Arts Block		
3:15	End of Day		

---

*Entries in the second column represent activities which take place outside of the observed space but do not involve all the students. Those students which remain in the space continue the activities in column 1. When an activity in column 2 is completed, the group returns and continues the activity specified in column 1.

THE ACTIVITY CONTROL BLOCK

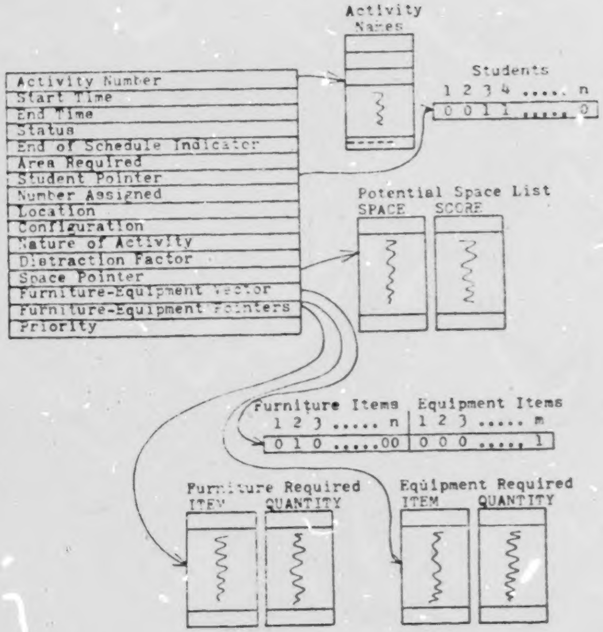

Activity Names
Students
1 2 3 4 ..... n
0 0 1 1 0
Activity Number
Start Time
End Time
Status
End of Schedule Indicator
Area Required
Student Pointer
Number Assigned
Location
Configuration
Nature of Activity
Distraction Factor
Space Pointer
Furniture-Equipment Vector
Furniture-Equipment Pointers
Priority
Potential Space List
SPACE
SCORE
Furniture Items
1 2 3 ..... n
0 1 0 00
Equipment Items
1 2 3 ..... m
0 0 0 1
Furniture Required
ITEM
QUANTITY
Equipment Required
ITEM
QUANTITY

Figure 3

Floorplan of One Pod of
an Observed Multiunit School

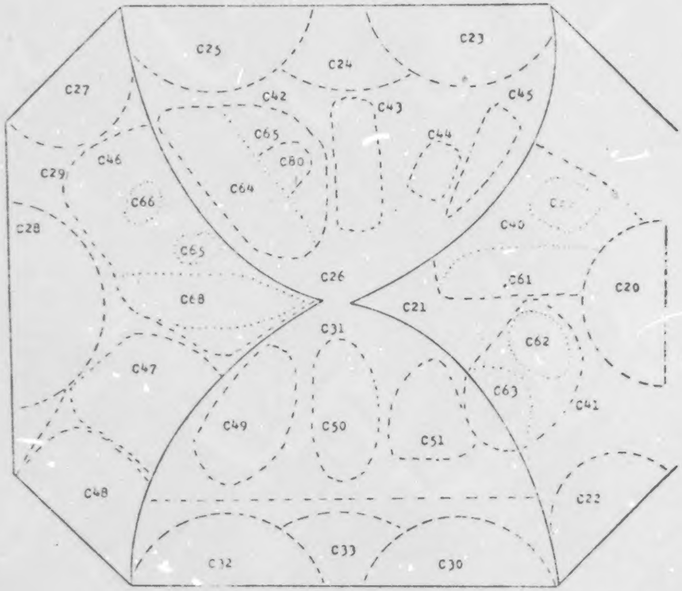

C25
C24
C23
C27
C42
C43
C45
C65
C44
C80
C46
C29
C64
C66
C40
C28
C65
C26
C61
C20
C68
C21
C31
C62
C47
C63
C41
C49
C50
C51
C48
C22
C32
C33
C30

Figure 4

Table 2

Summary of Space Use

Enrollment = 150

I SPACE	MODS IN USE	% MODS USED
C00	0	.00
C10	0	.00
C20	6	.25
C21	3	.13
C22	11	.46
C40	20	.83
C60	0	.00
C61	0	.00
C41	0	.00
C62	15	.63
C63	14	.58
C11	2	.08
C23	4	.17
C24	6	.25
C25	8	.33
C26	6	.25
C42	14	.58
C64	1	.04
C65	2	.08
C80	0	.00
C43	2	.08
C44	1	.04
C45	5	.21
C12	0	.00
C27	7	.29
C28	22	.92
C29	3	.13
C46	18	.75
C66	0	.00
C67	0	.00
C68	3	.13
C47	13	.54
C48	1	.04
C13	3	.13
C30	18	.75
C31	17	.71
C49	0	.00
C50	0	.00
C51	0	.00
C52	0	.00
C32	7	.29
C33	10	.42

Table 3

Summary of Space Use

Enrollment = 200

I SPACE	MODS IN USE	% MODS USED
C00	0	.00
C10	0	.00
C20	4	.17
C21	2	.08
C22	9	.38
C40	16	.67
C60	0	.00
C61	0	.00
C41	1	.04
C62	6	.25
C63	10	.42
C11	8	.33
C23	5	.21
C24	2	.08
C25	7	.29
C26	1	.04
C42	11	.46
C64	0	.00
C65	2	.08
C80	0	.00
C43	3	.13
C44	4	.17
C45	4	.17
C12	0	.00
C27	1	.04
C28	14	.58
C29	9	.38
C46	14	[illegible]
C66	0	.00
C67	0	.00
C68	0	.00
C47	4	.17
C48	0	.00
C13	1	.04
C30	21	.88
C31	11	.46
C49	2	.08
C50	0	.00
C51	1	.04
C52	1	.04
C32	4	.17
C33	5	.21

EXPLORING NEW WORLDS

Richard L. Ballard
Physics Computer Development Project
University of California
Irvine, California 92664

Electronic media have given us new eyes and ears. With them we can see into new dimensions, toward new frontiers, the outer worlds of space and the inner worlds of thought and knowledge. In these explorations, computers are at first our tools. Quickly they become the key, the door, and much that lies beyond. We know the beginnings, but not the boundaries of these new worlds.

As educators we accept in varying degree two roles: that of explorer--discovering and teaching ourselves--and that of guide--sharing what we and others learn, forging new paths. The shock of our future/present is that neither role may be neglected. The frontier is too diverse and advances much too fast to be lost sight of. As guides we must teach what is known, and more importantly, what will be known.

Access to new worlds today is very uneven. It varies from place to place, depending much on hardware, software, and expertise. Different disciplines find themselves more or less suited to immediate computer use. More in the case of mathematical sciences, less in the data base sciences, humanities, and arts. The views given here originate in a fortunate place, the University of California at Irvine, and represent considerable experience in a most mathematical subject, physics. They are thought, nonetheless, to be quite general.

FINDING NEW WORLDS

"New Worlds" is the title of one Physics Computer Development Project proposal given support by the National Science Foundation last year. It aims at developing breakthrough subjects, graphic communications, and intelligent natural language interaction. Collectively, these elements become literal play worlds wherein subject knowledge can be represented, manipulated, and made into new conceptual experiences. The casualness, immediacy, and general accessibility of these worlds makes complex and powerful ideas into toys for widespread use and experimentation. Teaching and concept testing in higher education can look to Piaget-like approaches. The strategies for doing this are based on firm successes with several existing educational programs.

Development of new conceptual worlds is perilous in its apparent over-choice. Computing can do many things--which should we support, given limited resources. For educators the temptation is to teach what we already know, to deal with new discovery later. Unfortunately all instructional development takes time, whether it is good or bad, useful or not. Schools lacking computers and/or specialized software readily view those as the critical barrier. Yet in places where all such things are available, the long haul of real development is clearly just begun.

Against today's slow development of effective teaching material, we have an explosive use of computers in subject research. As a medium, computers are transforming their subjects. In this situation, the impulse to use them to teach what is already known can guarantee instructional obsolescence. In physics, for example, many programs are being written to teach specialized analytic methods. Researchers, on the other hand, now make heavy use of more general numerical methods. These wait in the wings. Every expansion of computing that widens the use of existing educational programs invariably gives us the computing power to bring more relevant numerical methods into what we teach.

To survive very long, the new worlds we create on computers must be computer worlds. They must contain within them the unique computer methods which are most effectively changing the subject. To assure this, we should look to the computer breakthroughs, to areas in any discipline where old knowledge is encompassed and new knowledge is abundantly promised.

It may be possible to overdo in pursuing the future, of course. Today's view cannot be totally ignored. Colleagues and supporting agencies often look for immediate gains in processes they already understand. Good teachers have important things to say now, and the computer can be used to do it. We should seek good teaching everywhere. It is something else to ask computers to do it better, to revolutionize the way we teach. It is a good bet that for today's world the computer revolution may never come. Computers may never do what is done today more effectively, traditional subjects may always be taught in traditional ways. Proving otherwise may be non-productive and why bother. Every area that uses computers heavily for research does so for a reason, most often because there is no other way, there is no real competition. These are the new worlds, they already belong to the computer.

NEW WORLDS, WHAT WE MAY SEE

It is probably fair to say that no one has yet seen computer worlds as they will come to be seen. Arthur Clarke has called computers one of the great unanticipated inventions of history. None of our great dreamers in science fiction or elsewhere prepared us for computers. With all that has been written that is still the case. Our most absurd guesses find technical solution almost as soon as we guess them. Computers talking, seeing, thinking--whether you concede that these things are happening now or will happen soon depends as much on your definitions of these words as on the facts of computer behavior. If you believe that simple animals can do these things, it is increasingly difficult to deny them in computers.

These issues need never be settled. However, they do relate to what we expect as the near and distant futures of computing. Our choice of what we will try to do now is very much limited by assumptions, our inherent perception of what computing is and what it can become. In a sense, the New World Organization detailed below is a clear product of today's thinking, a pragmatic acceptance of today's resources. It exists now. It is very different from C.A.I. because we think computers are and should be something very different.

We expect that real computer worlds will be based on uniquely electronic forms of interaction. They won't be like books or like lectures, movies or TV. The future will say, "like computers," or some other word and know the difference. We can only guess, but some guesses are better than others. First, they will be graphic to an extent undreamed of. To those who have developed educational interactions in both words and pictures this is obvious and assumed. Educational programs without graphics were infinitely resistible; with graphics the arguments are few and far between. For authors, the transition is irreversible.

More powerful than experience are certain facts of biology. Iconic perception, the recognition and processing of visual patterns, is orders of magnitude more efficient for humans than other sensory means of communications. Between the parallel processing of pictures and the serial processing of words, there seems no competition. Having both is, of course, very nice indeed. Computers and vision are fundamentally matched, they share the same physical realm and limitation--the speed of light.

A second and related guess is that new computer worlds will appear as they must appear to make humans comfortable. Apart from transient hardware limitations, design will be dominated by psychology, human engineering, and the limits of programming skills and imagination. It is already clear that the harsh, domineering, "do not fold, spindle or mutilate" image of early computers has nothing to do with computers per se. Like today's language limitations, the feel of the interaction is restricted by the willingness of programmers to put themselves out for the user. Somewhere they draw the line, making their own lives easier, often citing the logical growth that comes to users from knowing at least that much "computereze." Fortunately, computer science builds upon itself. More accommodating programmers react to the user's natural desires; until, we presume, those desires will become the dominant specification.

If we consider what users may want, we find few universal models for a comfortable, interesting, and reasonably predictable world that can match the computer's potential complexity. One obvious possibility is a world that looks like another human being, perhaps in the guise of teacher, guide, or playmate. This choice is particularly efficient, assuming that both parties are intelligent. Contexts and meanings can be understood. As between humans, we can have large areas of mutual assumption. The conventions learned since birth through continuous human interaction can be extended to machines, commanding very complex responses without new or specialized skills. Still the computer's graphic expressiveness can clearly exceed a human's, other real world analogies will be necessary. Possibly it will be the view from a car or airplane, through a TV set, or ultimately a hybrid environment with the computer behaving both as a physical situation and as a set of explicit or implicit personalities.

CREATING NEW WORLDS NOW

Suppose one takes these views, goals, and speculations seriously and attempts to pragmatically implement them. Presumably even a substantial approximation to such a world could have value and much can be learned just trying. A kind of "proto-world" of this type was written by the author several years ago with the guidance and assistance of Alfred Bork. The result was called MOTION and has become an important experience for many students at Irvine, involving 1,000-3,000 student sessions per month. The enormous success of a program "that you just talk to" led to other experimental programs, like QUANTUM, and the New World programs now under development.

We have chosen to begin work in several subject areas: physics, chemistry, mathematics, art, and potentially in physiology, biology, and social science. Within these areas of interest several worlds are being created, often sharing key resources. The dominant activity is extraction of new graphic knowledge from mathematical bases. New representations and simulations are developed from the application of physical laws, mathematical functions, transforms, and operators. Other major activities include symbol manipulation in chemical and mathematical equations, aesthetic and symmetry group analysis of graphics, and the self-generation of data bases. More precise specifications for the initial worlds are developing as work begins.

The basic structures for these worlds have already been developed in MOTION, QUANTUM, GRAPH and other programs. The remainder of this paper will look at the details of that particular New World approach.

NEW WORLDS, MICROSTRUCTURE

Irvine's New Worlds consist of many small pieces. Individually many of these might be recognized by a research programmer or an author of computer-assisted instruction (C.A.I.). The elements most responsible for generating new knowledge are those labeled facilities or resources. Picked for their power and generality, each of these contains (1) a data structure, (2) numerous data manipulating subroutines, and (3) an analysis scheme that recognizes the pertinent data in every request for that facility. For lack of a better name, we call this latter recognition scheme, the "local intelligence," since it often looks at context and assumes a great deal of information for each use of a particular resource. This is a vital function in making the program human and interactive. Characteristically, activities that involve computational waits exceeding three to five seconds are shunned completely. This makes it possible to always do something. If the assumptions are inappropriate, users can then proceed to make themselves more explicit.

Facilities are inherently looping structures. They accept requests; they analyze them, setting subroutine and data base parameters; they modify or display information; and then they loop back to the start for a new request.

The second distinct microelement produces dialog with an author. The techniques of branched programmed instruction are often used here. This discussion is typically short, detailing a particular topic or idea. It supplies definitions, examples, test questions, and other informative elements. With a few important differences, discussions look very much like short, traditional, C.A.I. lessons. They represent the formal intervention of guides or instructors into the otherwise free play between users and facilities.

Discussion elements are first written into new worlds to provide minimal instructions on the use of facilities. Feedback expands them to further ease the interaction. If resources are well chosen, they must lead many user-authors to new discovery or to powerful examples of known phenomena. Author experience in a given world can be fed back, laying out productive paths, and reflecting the importance of particular choices.

A unique element in New World dialog is the root program or central intelligence. It is most responsible for the personality or outward appearance of the program. Following the QUANTUM model, New Worlds begin with two simultaneous presences: a conversational guide, who talks with text and animated graphics, and a graphic viewport that treats the program as a global map. Flying into the globe uncovers progressively deeper content maps. Ultimately these descriptors give way to participation in the labeled process.

The root's primary task is to respond intelligently to any unanticipated user statement. Every such situation is considered a possible attempt by the users to change the subject or ask questions of their own. Anything the world can do or or say, it will do whenever asked.

The central intelligence is responsible for moving users to requested information or facilities and for maintaining an intelligent conversation. It does this in ways, independent of the subject. What it knows of the content comes from vocabularies supplied with each facility and discussion. It possesses short and long-term memories and uses these in conversational protocols. It can repeat things, go back. It can suggest new topics. Important points are remembered and suggested whenever the user appears undecided.

NEW WORLDS, MACROSTRUCTURE

A flexible, organic environment is needed to accommodate expanding resources and continuous feedback from many author-explorers. Typically, the user is in immediate contact with only a small active portion of this world. The greatest bulk of any world can lie dormant until called. There are many hardware/software configurations that permit this. The active part may be a mini or a very intelligent terminal. It may be the in-core time slice of a larger timeshared computer. The dormant mass may be the single master copy stored on magnetic or video disc, locally or at some distant site. New Worlds develop information either about the user or for his future use or for modification and growth of the program itself. This requires volatile space, organized by program and by user.

Irvine's proto-worlds are written for a timeshared computer. Complete load modules exist on discs. Typically they are several hundred thousands of words long, combining DIALOG (graphic and language analysis macros) and FORTRAN (numeric routines). This mass is highly structured in an expandable set of overlays. At any given time the overlay pattern in memory is a small fraction of the total. We try to keep this ready component at 20-25K words or less. Larger partitions, while possible, usually face discrimination by systems of time and resource sharing. The overlays in memory furnish discussion, facilities, and a vocabulary for dealing with the present topic, they can also pull in related topics.

The central intelligence possesses a simple keyword set that it uses to detect complete change in subject. To go further, it swaps (overlays) into memory the appropriate subject vocabulary. If it has repeated difficulty, it reorganizes itself to include the entire world vocabulary at the temporary loss of previous content.

The purpose of the vocabularies is to provide the central intelligence with descriptors of each topic and with simple keyword tests to sort out the most appropriate response to any input. Initially only simple sorting is needed. Computer dialog is seldom limited by its ability to understand user input, more by the willingness of authors to provide many responses.

Once the most appropriate topic has been brought into core, more detailed syntax analysis can be performed. In the case of very powerful facilities, analysis may involve parsing (compiling formulas into machine code) or elaborate symbol translations.

A BEGINNING

New Worlds create new educational opportunities using powerful methods unique to the computer. They are not merely computers assisting instruction nor instruction in computer methods. They are an interactive result, more than both and different from either. Their creation depends upon both cooperation and division of labor between educators, content specialists, and computer scientists.

Invariably our most successful developments differ markedly from what we began with. We expect that New Worlds will assume their own forms through use and feedback. Development is organic and evolutionary rather than the final fulfillment of some grand and predetermined plan. We start with seed worlds at Irvine. Potential world developers break off the subject free root and certain dialog facilities from existing programs. This structure is a complete dialog. It can talk to the author; it knows about dialog writing, but nothing about its future content. It embodies the experience and combined efforts of previous developers, but it can be changed in part or "in toto." It is just a beginning.

The Use of Computer Modulated Drawing in the Teaching of Art

Charles M. Williams

Computer Science Department
Virginia Polytechnic Institute
and State University

The figures shown in this paper are examples of "Computer Modulated Drawings". They were derived from a single source drawing and were produced by software developed by Williams [1] to allow artists to use a computer as a tool in rendering their own work. The process uses a Visicon Automatic Digitizer [2] to convert a hardcopy drawing into digits which are then manipulated by graphic collation software [3,4] into data structures which isolate and mathematically describe the geometry of the various lines which compose it. These data structures may be used as a direct source of commands for a digital plotter. Modulation occurs when the mathematical descriptions themselves are dynamically perturbed at the time of plot generation.

The simplest form of modulated art is no modulation at all, and the results are a faithful pen and ink reproduction of the original drawing. Heavier modulation results in a work which retains the general holistic form of the original but which has been rendered in a completely different style. The process is totally automatic and rapid, requiring (for the figures shown) roughly two minutes for digitizing, four minutes for graphic collation (on an IBM 370/158) and twenty minutes for plotting. Graphic collation times are roughly linearly proportional to the amount of line data present, and plotting times vary with the complexity of modulation. The entire system is parameterized and does not require a knowledge of computer programming. Each type of modulation is coded by number, and the degree of perturbation is specified by such variables as amplitude and spatial frequency.

The value of computer modulated drawing to the teacher of art is derived from a capability for allowing a drawing to serve as a single and constant source of data for an indefinite number of variants to it. This permits the detailed and parameterized study of the visual effects achieved by varying line structure, line texture, pen stroking style, etc. in rendering truly constant subject material. Such renditions are not possible by hand, for subtle localized changes in style will inevitably be introduced by the human while redrawing subjective material.

In essence the computer modulated drawing process provides the art teacher with the facilities of a computer darkroom with which to study the effects of different drawing styles on various subject matter. The source data for these studies may be carefully chosen archival works or they may be spontaneously selected drawings from the class itself. The speed of the computer based process insures that results will rapidly be forthcoming even in the case of the most complex and tediously structured design patterns.

The times required by computer modulation are currently largely a function of plotter speeds. Consequently the substitution of a display scope for a mechanical plotter can yield faster, though perhaps visually degraded, results. The plotter, of course, can manually be controlled in its employment of ink color and pen widths. The same data may be repeatedly plotted to yield high quality results on differing grades of paper.

The computer modulation process was expressly designed to give the artist the benefits of a computer while allowing him to perform much of his work in his accustomed environment unfettered by the mechanical and electronic constraints often associated with machines. He prepares his work at his leisure using standard art supplies of his choice. When he has finished he secures his drawing to the surface of a drum which is mechanically rotated past an optical sensing mechanism which records black and white decisions based on the amount of light reflected from the document. Placement of the drawing on the drum requires roughly one half minute and needs no special mechanical aptitude. Digitization proceeds automatically and is completed in at most two minutes.

The digitizer output is fed to a computer which then creates the geometrically descriptive data structures. Modulation begins when the modulation parameters have been specified. The results are plotted mechanically on paper or electronically on the face of a cathode ray tube.

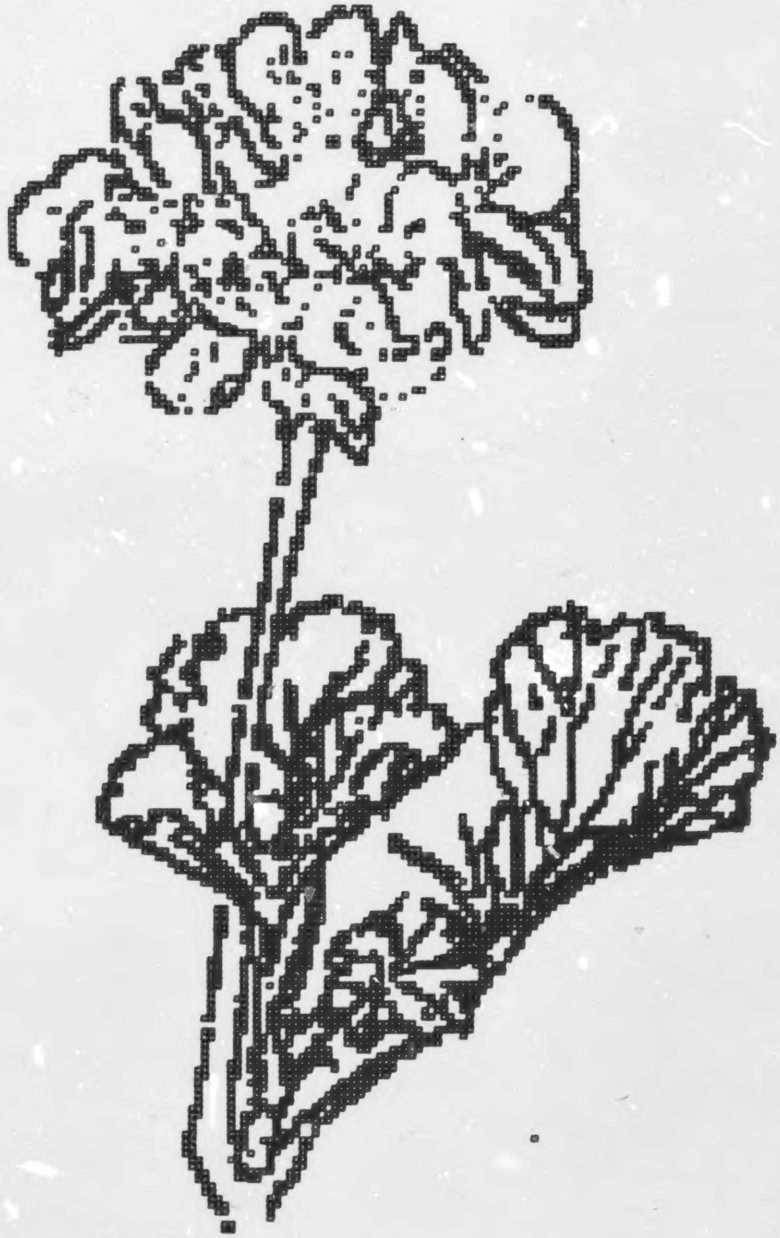
A typical hardware configuration may include a digitizer, incremental plotter, and display scope connected to a minicomputer which can communicate with a larger general purpose computer. The current graphic collation and modulated drawing software system resides in 160K bytes of IBM 370/158 memory but hopefully future systems will totally be resident within the minicomputer itself. It is envisioned that future systems will incorporate computer graphics facilities which will allow an artist to manipulate and modify his drawing interactively.

The art teacher can employ such equipment throughout his domain of expertise in attacking problems in drawing and, hence, in the photomechanical print (etching, lithograph, and silk screen). Beginning students can study the effects of style on rendering simple subjects, and advanced ones can selectively incorporate widely varying modulation techniques in creating arbitrarily complex works. The full power of computer graphics can be invoked to assure that pleasing localized effects on a drawing are not destroyed while others are being developed elsewhere.

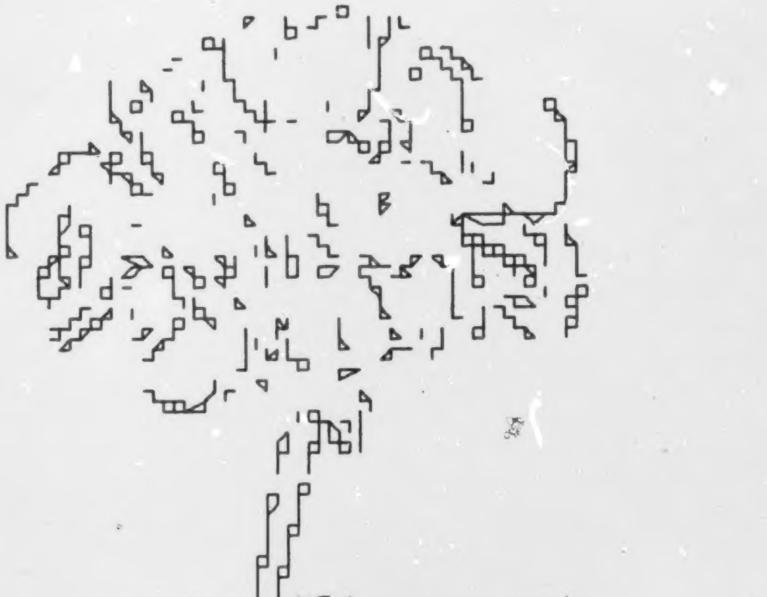
In short the computer modulated drawing process provides the teacher of art with a new and exciting tool with which to work. The computer is involved in its appropriate domain of tedium, and the artist retains his dominance in creativity. Though the system is mechanical, the results need not have a mechanical appearance.

References

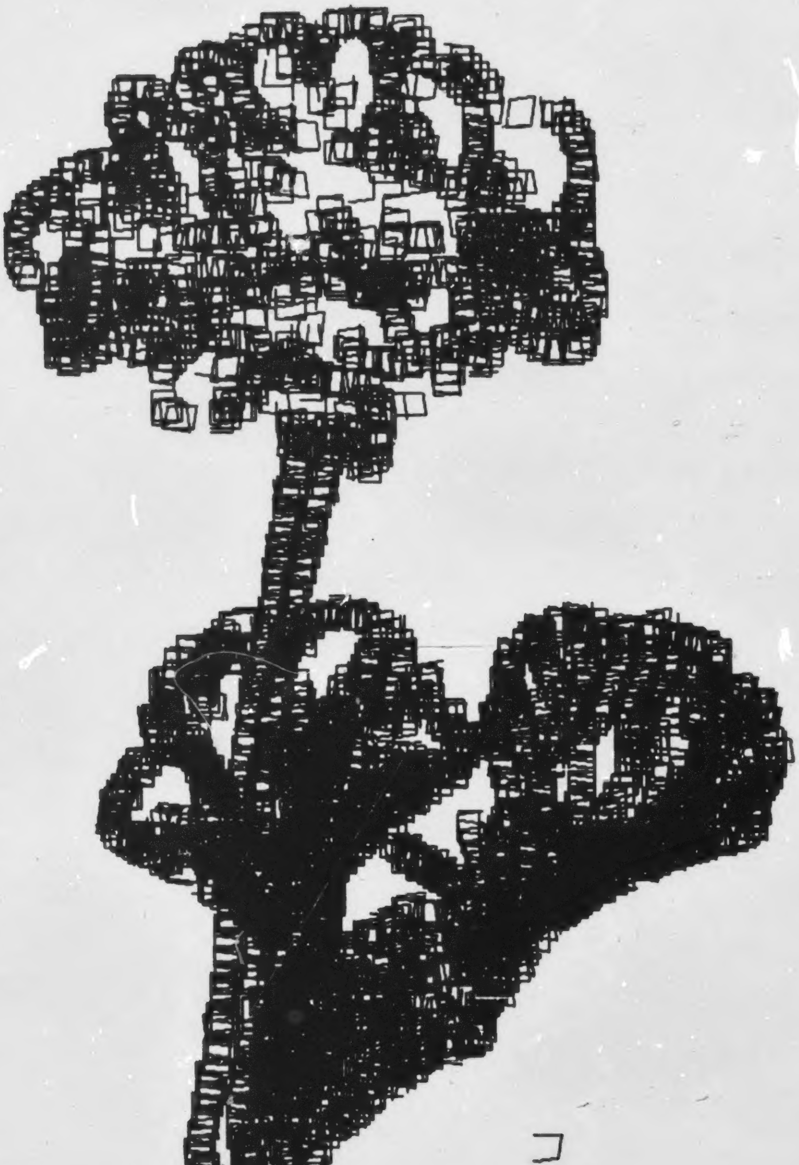
1. Williams, C.M., Sarvis, A.T., "Computer Modulated Drawing", Southeastern College Art Conference, Chapel Hill, North Carolina (November 1974).
2. Broomall Industries, Inc., 682 Parkway, Broomall, Pennsylvania 19008, U.S.A.
3. Williams, C.M., 'The Automatic Transduction of Drawings into Data Bases', Proceedings of AFIPS 1973 National Computer Conference (June 1973), pp. 635-637.
4. Williams, C.M., 'The Role of Automatic Digitizers in Computer Aided Design', Proceedings of CAD74 International Conference on Computers in Engineering and Building Design, Imperial College, London (September 1974), IPC Science and Technology Press Ltd.
5. Myers, Eugene D., Computer Science Department, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24061, U.S.A.



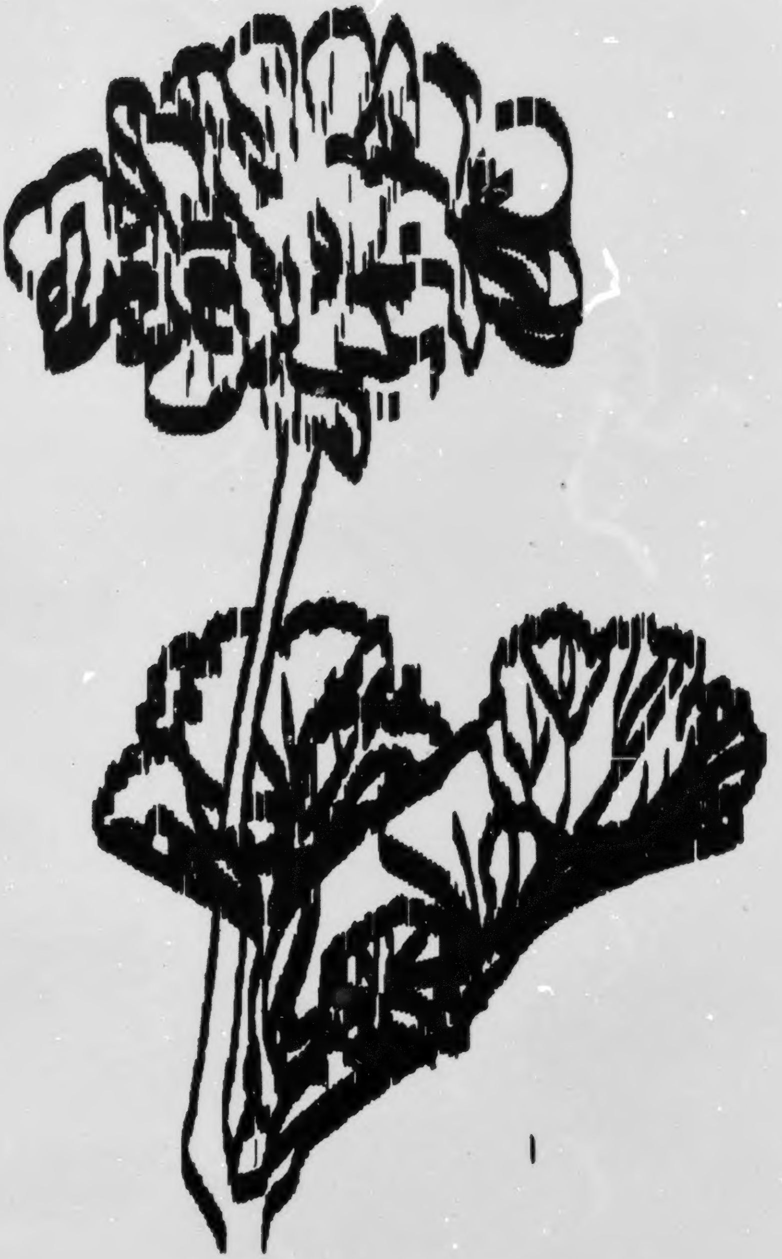
Flower modulated by a beaded pattern
created by a mosaic of small squares



Flower modulated by a "needlepoint" technique
involving gridding on a coarse mesh



Flower modulated by a Myers [5] box pattern consisting of overlapping deformed rectangles



Flower modulated by a "woodcut" technique using Myers box patterns of zero width

SUMMARY OF
TICCIT PANEL

The first complete instructional use of The MITRE Corporation's TICCIT CAI system will be discussed by the TICCIT project directors at Phoenix College, Phoenix, Arizona and the Alexandria Campus of Northern Virginia Community College. New TICCIT applications in special education for the handicapped and new TICCIT configurations will also be described.

Participants:

Mr. Dunstan Hayden
Northern Virginia Community College

Mr. Fred Morrison
Phoenix College

Ms. Wanda Rappaport
The MITRE Corporation

"PRIVACY AND THE PRUDENT COMPUTER-MAN"

David Allen Weinberg
Computer Audit Systems, Inc.

The future is rapidly crossing the threshold of 1984, but with unexpected anti-Orwellian consequences. We, in data processing, are fearful of maintaining and processing freely acquired data. This is data from individuals who have usually requested that such material be made available to a specific institution for purposes of supporting the individual's future aspirations.

What we are realizing is that by 1984 our primary fear may be of our inability to utilize openly disseminated information. This acutely contrasts the supposed Orwellian nightmare of hidden governmental data gathering and surveillance activities that destroy the individual citizen's right to privacy.

As this time it is difficult to be certain of the exact flavors and shadings of the future legislation which will assuredly be forthcoming. What is required today are not prophesies of the future, but patterns for positive design and action.

What is the heart of our problem today? I do not share the conviction that we must fear criticism or even legal actions for failing to comply with standards and laws which are presently undefined. I do believe that today, as DP administrators, we are expected to take such precautions as can reasonably be expected of a competent professional.

We must act based on what a "prudent man" would be expected to know and do. This list of knowledge is quite broad. It does not specifically require intimate familiarity with the "Privacy Act of 1974" (more properly S 3418). Neither is the prudent man expected to meet all the criteria contained in such texts as the AFIPS, "Security Systems Review Manual"; or FIPS, "guidelines for Automatic Data Processing, Physical Security and Risk Management".

However, a "prudent man" would be expected to be familiar with these documents. He should also be in a position to demonstrate that, based on practical constraints, his DP center is complying with "reasonable data security and legislative expectations"; or, that as a minimum, the DP center has plans which will result in such compliance.

Your role for the future should be active. Begin taking those steps to safeguard data --- steps that a reasonable man would take. And, become active in influencing future legislation. Let your governmental servants hear from you. Let them know what the real world is like. The collective voice of thousands of prudent men should not go unheard.

The task before you is not easy. However, as professionals, it is within your grasp. We should all look forward to 1984 with confidence.

MODEL-BUILDING AND COMPUTER SIMULATION FOR NON-PROGRAMMING USERS*

William L. Bewley

Minnesota Educational Computing Consortium
Lauderdale, Minnesota 55113

> "How strange... that 'computers in education' should so often reduce to 'using bright new gadgets to teach the same old stuff in thinly disguised versions of the same old way'."
>
> Papert and Solomon, 1972

This paper describes progress toward developing a technique that may help us use the computer to teach new stuff in a new way. The technique is based on the idea that learning involves building internal (mental) models of reality, testing these models against reality, and correcting (debugging) the models when errors are found. The idea is not new--it has been expressed by Minsky (1968, 1970) and its philosophical roots lie in the Cartesianism of the 17th century. What is new is the technique--the digital computer is used to help the user externally represent, test, and debug his internal models.

The technique assumes no user knowledge of computer programming. The computer is programmed to perform several information-processing operations appropriate for a certain class of models, e.g., queueing models. The user builds a model by specifying the sequence in which the operations are to be performed. The program then runs a simulation of the model. Because the user has not written a computer program, any errors revealed by the simulation can be attributed to the logic of the model. The user debugs the model by changing the sequence of operations.

Two sample model-building programs, Q and EAT, are described below.

Program Q--A Queueing Model Program

Table 1 describes operations appropriate for queueing models, which are provided by the BASIC program called Q. There are two kinds of operations required to build a model: actions (which do things) and questions (which ask if things have been done). In Table 1, operations 1 through 6 are the actions and operations 7 through 11 are the questions. There are two operations in the table that are neither actions nor questions--operation 0 (zero) which is used in inputting the model, and operation 12, which is used to obtain information on the current status of model elements.

A generalized flowchart for program Q is shown in Figure 1. The program begins by asking the user to input his model by specifying the order in which operations (identified by the numbers in Table 1) are to be performed. When the number for an action is entered, the program asks for the next operation to be performed. When the number for a question is entered, the program asks for two operations: (1) the next operation to be performed if the answer is yes (the "yes" branch) and (2) the next operation to be performed if the answer is no (the "no" branch). The program then follows the "yes" branch leading from the question, asking for the operation to be performed following the "yes" answer. It continues asking for operations until a 0 (zero) is entered as an operation number; this means that there are no more operations to be entered. At this point, the program checks to see if it has followed the "no" branch from every previously specified question. If it hasn't, it goes back and begins asking for operations in the most recently specified "no" branch which needs to be followed. If it has exhausted the "no" branches, it stops asking for operations and lists its representation of the flowchart. A sample run of program Q showing the input of a model is given in Figure 2. The model is represented as a flowchart at the left of the figure.

The trickiest part of entering a flowchart is in forming loops. A loop occurs whenever a series of at least two operations is to be performed again. The program recognizes a loop whenever the first two operations in the repeated series (including both branches of a question) are entered and then followed by a zero entry. Thus, to form the loop from "Queue Empty" to "Input a User," operation 1 is entered, then operation 7 with both of its branches, and then 0.

As the user enters a model, the program builds a flowchart. The program's flowchart, a four-column matrix of numbers, is printed when the user has finished entering the model. The first column of the matrix numbers each row. The second column contains operation numbers. The operation in row 1 is the first to be performed. The third and fourth columns contain the row number of the next operation to be performed. If the operation which has just been performed is an action, the row number for the next operation is in the "yes" column.

---

* The programs described in this paper were developed with support from the National Institute of Education while the author was with the Computer Technology Program of the Northwest Regional Educational Laboratory, Portland, Oregon.

Thus, if we start at row 1 and perform operation 1, the next operation is the operation at row 2, operation 7. The "no" column contains a non-zero value only in rows containing a question. In those rows, the number in the "yes" column is the row at which the operation following a "yes" outcome is to be found, and the number in the "no" column is the row at which the operation following a "no" outcome is to be found.

After it has printed the flowchart, the program asks the user to input values for independent variables and the desired number of iterations for a simulation of the model. The sequencing of operations for the simulation is controlled by a computed GOTO statement which transfers control to the segment of the program corresponding to the operation indicated by the flowchart. The simulation continues until an action leads to an error (see the example in Figure 2) or the desired number of iterations has been reached. When the simulation ends, the program is either stopped or continued as shown in Figure 1.

Program EAT--An Energy Consumption Program

Table 2 describes the operation appropriate for program EAT, a BASIC program having the same generalized flowchart as program Q, but testing models of a different class. Operations 1 through 5 are actions; operations 6 and 7 are questions. Operations 0 (zero) and 8 through 10 are special-purpose operations: operation 0 (zero) closes loops and operations 8 through 10 output the current status of model elements.

The class of models handled by EAT are simple energy consumption models. The system to be modelled is the biological system controlling hunger and eating behavior in an imaginary animal (see Lindsay and Norman, 1972, pp. 595-604). The inputs to the system are the number of hours in the observation period (how long we want to watch the animal), the initial energy reserve (the rese reserve of energy present at the beginning of the observation period), the goal energy reserve (the energy reserve needed to maintain life), actual incentive value (the incentive value of food available in the environment), the goal incentive value (the incentive value of food that the organism would like to eat), the energy consumption rate (the amount of the energy reserve consumed during one hour of the observation period), and the energy intake rate (the amount of energy generated from the food eaten in one hour of the observation period). These inputs are, of course, the independent variables. The actual values assigned to these variables are unimportant; the concern is with the relation of the value of one variable to the value of another, e.g., whether the intial energy reserve is greater than, equal to or less than the goal energy reserve. The outputs of the system are three behaviors: searching for food, eating food, and sleeping. A sample run of Program EAT is shown in Figure 3.

Final Comments

Program Q and EAT have been described in order to show how the computer can be used to teach new stuff in a new way. The new stuff is the building and debugging of models. The new way is the use of the digital computer as a medium in which to do the building and debugging. There have been other attempts to put model-building into the curriculum; the Man Made World of the Engineering Concepts Curriculum Project, Polytechnic Institute of Brooklyn, involves the use of a special-purpose miniature computer by high school students; William Dorn of the University of Denver teaches math to college students by having them write their own computer simulations; and Seymour Papert and the LOGO group at M.I.T. have taught math to elementary school students using the LOGO language and a programmable "turtle."* The present approach differs from these, however, in that there is no need for special equipment like a miniature computer or a turtle, for a special-purpose language like LOGO, or for assuming that students know how to program. All that is needed is a computer, a teletypewriter terminal (assuming interactive programs, though fast batch is certainly possible), any general-purpose language, and at least one person (the instructor, another faculty member, or a student) who can program. Programs like Q and EAT can be created for any class of models by defining the actions and questions to be provided and following the generalized flowchart of Figure 1.

Our work is still in the developmental stage. No full-scale testing has been done, but the high school students who have used the programs enjoy them and seem to be able to use them. One source of difficulty that we have found is that the procedure for forming loops is awkward and sometimes confusing. We are attempting to remedy this problem by allowing the user to enter the word "LOOP" followed by the operation number at which the loop is to end. We also intend to develop programs appropriate for other classes of models. We are especially interested in developing more realistic energy consumption systems which allow the use of limited energy sources and include the need to expend some available energy to produce more energy.

References

Lindsay, P.H., and Norman, D. A. Human Information Processing. N.Y.: Academic Press, 1972.

Minsky, M. Matter, Mind, and Models. In M. Minsky (Ed.), Semantic Information Processing. Cambridge, Mass.: The MIT Press, 1968.

Minsky, M. Form and Content in Computer Science. Journal of the Asso. for Computing Machinery, 1970, 17(2).

Papert, S., and Solomon, C. Twenty Things to Do With a Computer. Educational Technology, 1972, 12(4), 9-18.

---

* Although Papert's work is usually considered "problem solving" rather than "modeling and simulation," the students are externalizing their internal models in giving instructions to the turtle.

Table 1

The Operations Performed by Program Q.

Operation Number	Name	Description
0	End Loop	Used in forming loops. Tells the program that a loop has been completed.
1	Input a User	Gets next user from pool of potential users not in queue or service facility
2	Enter Queue	Puts the user obtained in Operation 1 into queue
3	Enter Facility	Puts user at top of queue into the service facility
4	Leave Facility	Outputs user in service facility
5	Advance Queue	Moves each user in the queue up one place in the queue, i.e., second in line becomes first in line, etc.
6	Increment Time	Adds 1 to current time; stops program when run time is reached
7	Need Service?	Determines if user obtained in Operation 1 needs service; prints user number if yes
8	More Users?	Determines if all potential users in pool have been obtained by Operation 1
9	Facility Empty?	Determines if there is a user in the service facility
10	Service Done?	Determines if service on the user in the facility has been completed; prints user number if yes.
11	Queue Empty?	Determines if there are any users in the queue
12	Output Current Status of Model Elements	Causes printing of current contents of service facility and queue

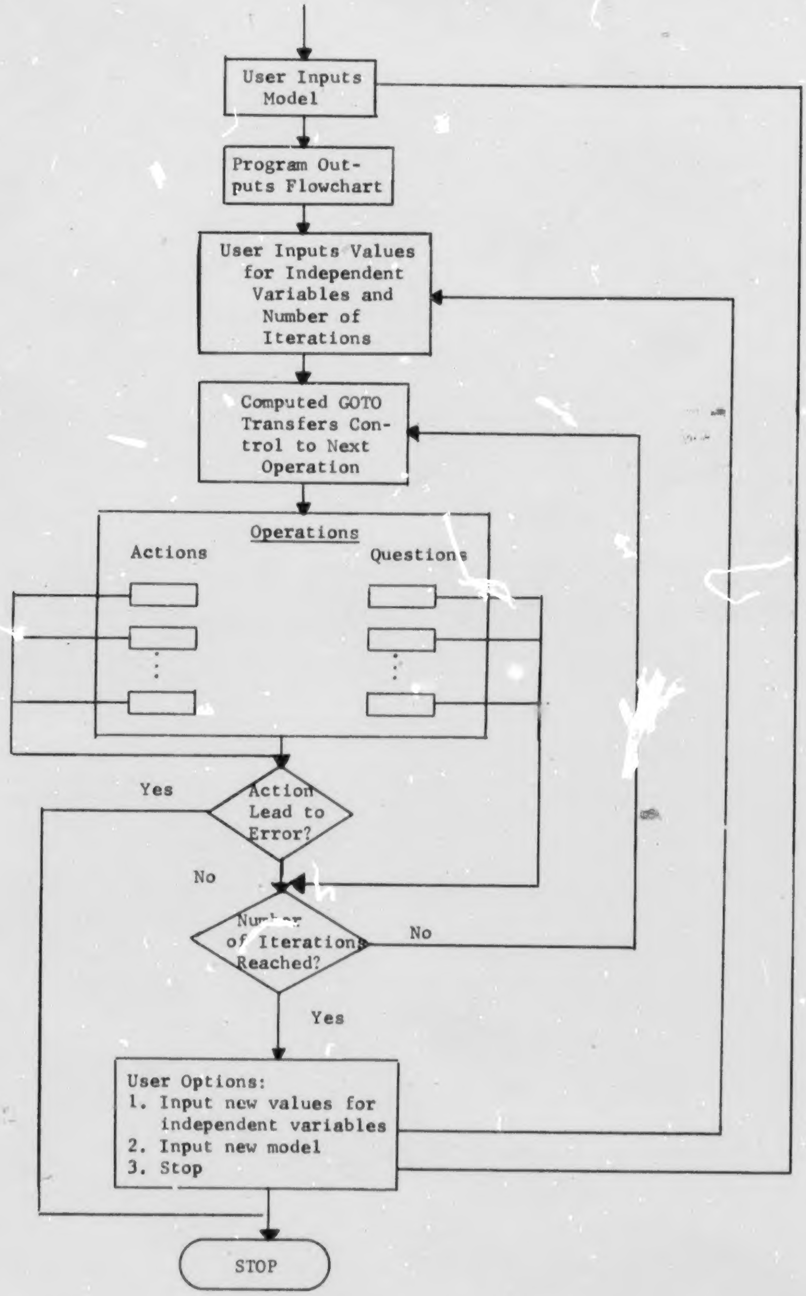
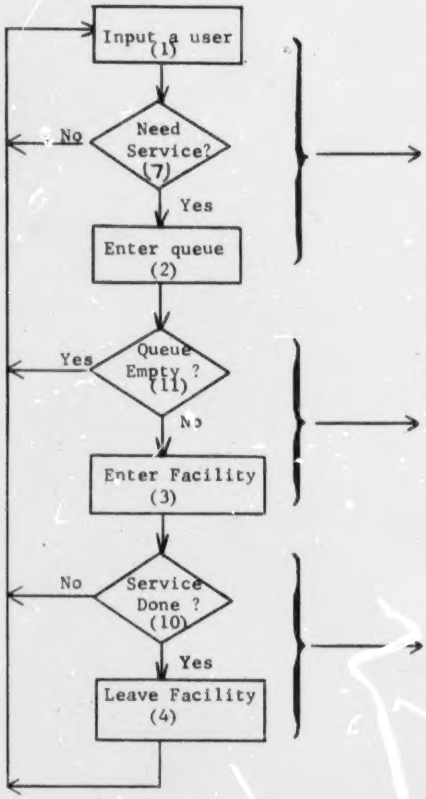

User Inputs Model
Program Outputs Flowchart
User Inputs Values for Independent Variables and Number of Iterations
Computed GOTO Transfers Control to Next Operation
Operations
Actions
Questions
Yes
Action Lead to Error?
No
Number of Iterations Reached?
No
Yes
User Options:
1. Input new values for independent variables
2. Input new model
3. Stop
STOP

Figure 1. A Generalized Flowchart for Program Q.



```
INPUT FIRST OPERATION NUMBER?1
NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
```

```
STARTING FROM NUMBER 2     NEXT NUMBER?11
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?1
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?3
STARTING FROM NUMBER 1     NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Queue Empty" to "Input a User"

```
GOING BACK TO YOUR LAST USE OF NUMBER 11    YOU SAID
THAT IF THE OUTCOME IS 'NO' THE NEXT NUMBER
WOULD BE 3
STARTING FROM NUMBER 3     NEXT NUMBER?10
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?4
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
STARTING FROM NUMBER 4     NEXT NUMBER?1
NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Leave Facility" to "Input a User"

```
GOING BACK TO YOUR LAST USE OF NUMBER 10    YOU SAID
THAT IF THE OUTCOME IS 'NO' THE NEXT NUMBER
WOULD BE 1
STARTING FROM NUMBER 1     NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Service Done to "Input a User"

```
GOING BACK TO YOUR LAST USE OF NUMBER 7    YOU SAID
THAT IF THE OUTCOME IS 'NO' THE NEXT NUMBER
WOULD BE 1
STARTING FROM NUMBER 1     NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?1
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Need Service" to "Input a User"

```
                        FLOW CHART
ROW            OPERATION          YES                NO
 1                1                2                  0
 2                7                3                  1
 3                2                4                  0
 4               11                1                  5
 5                3                6                  0
 6               10                7                  1
 7                4                1                  0


INPUT VALUES FOR INDEPENDENT VARIABLES
RUN TIME?5
NUMBER OF POTENTIAL USERS?5
PROBABILITY A USER NEEDS SERVICE IN A TIME PERIOD?1
SERVICE DURATION?2


USER 1     NEEDS SERVICE

USER 2     NEEDS SERVICE

ERROR AT OPERATION 3!
AT TIME 1     THE FACILITY WAS FILLED BEFORE IT HAD
              BEEN EMPTIED
DONE
```

Figure 2.

A Sample Run of Program Q.

The model contains several bugs.
This run reveals one of them.

Table 2

The Operations Performed by Program EAT

Number	Name	Description
0	End Loop	Tells the program that a loop has been completed.
1	Need	Need = Goal Energy Reserve - Current Energy Reserve*; prints new value
2	Current Energy Reserve I	Current Energy Reserve = Current Energy Reserve - Energy Consumption
3	Current Energy Reserve II	Current Energy Reserve = Current Energy Reserve - Energy Consumption + Energy Intake
4	Desirability	Desirability = Actual Incentive - Goal Incentive; prints new value
5	Excitation	Excitation = Need + Desirability
6	Need ≤ Zero?	Determines if need is less than or equal to zero.
7	Excitation ≥ Zero?	Determines if Excitation is greater than or equal to zero.
8	Sleep	Outputs hour and the word "Sleep;" increments time
9	Eat	Outputs hour and the word "Eat;" increments time
10	Search	Outputs hour and the word "Search;" increments time.

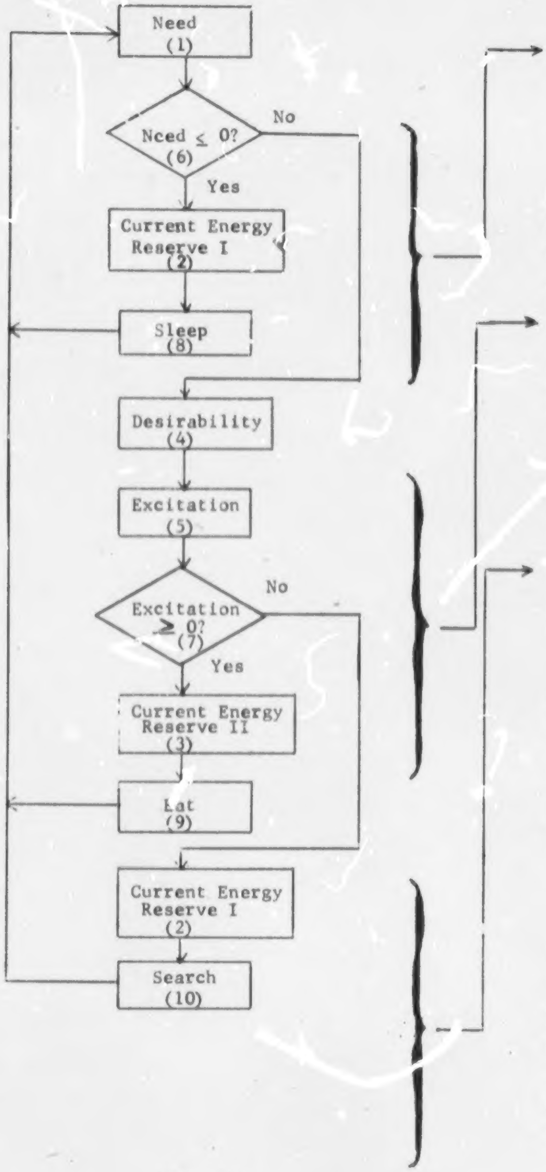


Figure 3.

A Sample Run of Program EAT.

```
INPUT FIRST OPERATION NUMBER?1
NEXT NUMBER?6
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?4
STARTING FROM NUMBER 2     NEXT NUMBER?8
NEXT NUMBER?1
NEXT NUMBER?6
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?4
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Sleep" to "Need"

```
GOING BACK TO YOUR LAST USE OF NUMBER 6     YOU SAID
THAT IF THE OUTCOME IS 'NO' THE NEXT NUMBER
WOULD BE 4     .
STARTING FROM NUMBER .4              NEXT NUMBER?5
NEXT NUMBER?7
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?3
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?2
STARTING FROM NUMBER 3     NEXT NUMBER?9
NEXT NUMBER?1
NEXT NUMBER?6
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?4
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Eat" to "Need"

```
GOING BACK TO YOUR LAST USE OF NUMBER 7     YOU SAID
THAT IF THE OUTCOME IS 'NO' THE NEXT NUMBER
WOULD BE 2     .
STARTING FROM NUMBER .2              NEXT NUMBER?10
NEXT NUMBER?1
NEXT NUMBER?6
INPUT NEXT NUMBER IF OUTCOME IS 'YES'?2
INPUT NEXT NUMBER IF OUTCOME IS 'NO'?4
STARTING FROM NUMBER 2     NEXT NUMBER?0
```

Forming Loop from "Search" to "Need"

```
                      FLOW CHART
ROW             OPERATION       YES              NO
 1                1              2               0
 2                6              3               5
 3                2              4               0
 4                8              1               0
 5                4              6               0
 6                5              7               0
 7                7              8               10
 8                3              9               0
 9                9              1               0
 10               2              11              0
 11               10             1               0


INPUT VALUES FOR INDEPENDENT VARIABLES
HOURS IN OBS. PERIOD?4
INITIAL ENERGY RESERVE?98
GOAL ENERGY RESERVE?100
ACTUAL INCENTIVE VALUE?23
GOAL INCENTIVE VALUE?25
ENERGY CONSUMPTION RATE?3
ENERGY INTAKE RATE?4


NEED= 2
DESIRABILITY=-2
HOUR 1     SEARCH


NEED= 5
DESIRABILITY=-2
HOUR 2     EAT


NEED= 4
DESIRABILITY=-2
HOUR 3     EAT


NEED= 3
DESIRABILITY=-2
HOUR 4     EAT
WANT TO CHANGE VALUES FOR INDEPENDENT VARIABLES?NO
WANT TO RUN IT AGAIN WITH THE SAME VALUES?NO

DONE
```

USE OF COMPUTER SIMULATIONS
IN LOGISTICS MANAGEMENT EDUCATION

David A. Ameen, Ph.D.
Virginia Commonwealth University

On the ground floor of a three-story, reinforced concrete and cinderblock building, located approximately 135 miles south of the Pentagon, a group of field grade officers and senior Department of the Army civilians huddled closely around a table in a brightly lighted conference room. The site of this meeting was near the site of a great Civil War battle--the Battle of the Crater, Petersburg, Virginia. This group had a problem similar to that of the logisticians of the northern and southern forces during the battle for Petersburg--the mission of supplying field forces with the necessary material to support the combat mission. Of course, the task was magnified tremendously; this was the nuclear age, and the implements of war ranged from nuclear weapons and missiles to the individual weapons required by the modern soldier.

These senior army personnel were discussing funding, budgeting, main battle tanks, overhaul scheduling, helicopters, generators to support missile systems, track shoes for tanks, paint, and some nuts and bolts. Their interest and enthusiasm were real and they freely interchanged thoughts, ideas and recommendations.

The members of this group were students attending the U.S. Army's Logistics Executive Development Course at Fort Lee, Virginia, and were actively engaged in a computer-assisted, decision-making logistics exercise known as CALOGSIM. This is but one of several courses conducted by the United States Army Logistics Management Center at Fort Lee, Virginia. The center's mission is to develop management skills of both middle and top level army logistics managers and provide uniform training in United States Army wide aspects of logistics management. This training is required to fill a critical need for adequately trained management personnel in the wholesale logistics system.

The case method of instruction used so successfully by many graduate level business schools was initially selected as the principal method of instruction at the center. The study and analysis of real or fictional cases provides the student with an insight and understanding that is normally acquired only through years of experience. The interchange of thoughts between members of the group, the give and take of lively discussion, and sound logical decision making provide the ingredients for excellent learning. Frequently, however, conclusions reached or decisions made cannot be tested against the situation; they can only be compared against history or left unresolved. In situations where many complex variables are involved this leads to some frustration since the student cannot live with his decisions or see the results of his actions. To overcome this shortcoming of the case method of instruction and to emphasize learning by doing, the use of computer simulations and business gaming was explored.

A computer simulation used for training may be considered to be an extension of the case method, primarily in two major areas.

The first area concerns student involvement. In a case a student analyzes a situation and attempts to project himself into the roles of the managers depicted in the case. In effect, he might say that had he been Colonel X he would have taken a different course of action or reacted in a different manner; but, had he really been Colonel X he may have reacted in much the same way as Colonel X did. In other words, in studying a case, the student looks at a situation in an objective rather than a subjective manner; whereas, in a simulation he is subjectively involved.

The second area concerns living with decisions. Going back to the previous illustration in which the student stated that had he been Colonel X he would have taken a certain course of action, another student might counter with the statement that had he taken that course of action, six months from now he would be in trouble due to the effects of certain probable events which might occur in the future. However, in a system as complex as the army's logistics system, with the many interacting variables involved, it is almost impossible to predict the effects of these variables on the student's decisions. However, with the use of a computer to simulate the passage of time, the effects of interacting variables on a particular decision can be readily measured. So the student really is forced to live with his decisions.

One of the primary objectives of the center is to train students in effective supply management. In support of this objective, the question

123

was asked, "What really makes one man a more effective supply manager than another?" It was found that in addition to having the necessary basic attributes of intelligence, flexibility, adeptness, etc., that what made one man a better manager than another was generally years of experience. The question was how could we give the students years of experience in the few weeks they were in attendance at our courses. The answer we found, was to use computer simulations to compress time; thereby, giving students so-called accelerated experience.

Computer simulations are used in nearly all the center's courses. They generally occur at the end of the course serving as a more or less capstone of the course. It is at this point that the students have an opportunity to put into practice what they have been taught.

Described below are three of the center's major computer simulations. All of the simulations have a scenario programmed into the computer which simulates the passage of time and the effects of both student decisions as well as certain real life events which occur in the particular model concerned.

CALOGSIM (A Computer Assisted Logistics Simulation)

CALOGSIM simulates the functions of a commodity management group at an Army National Inventory Control Point. The functions that the simulated commodity management groups perform during the simulation are typical of those performed at a real-life National Inventory Control Point. They are as follows:

Requirements determination
Procurement direction
Distribution management
Rebuild direction
Disposal direction
Controls
Decision making

The supply management group manages both principal items and minor secondary items and repair parts. The data pertaining to these items are based upon real-life data collected from actual NICP's. The simulation is designed to cover the management of fifty principal items and 450 minor secondary items and repair parts. The group manages the principal items by exercising worldwide control and develop materiel programs to fund their procurement. The problem of phasing in new items, geographic standardization, repair scheduling, redistribution of stock, initial provisioning and funding face the principal item managers. These problems become more real as the shortage of funds restricts inventory modernization.

The minor secondary items and repair parts range from nuts and bolts to large assemblies. Many of the items support the principal items and maximum interchange of information is required to manage the items properly.

The supply management group controls a distribution system consisting of eight U.S. Army depots. The demands placed on the system are controlled by a scenario and simulates requests placed by customers. To fulfill the demands, the managers take appropriate action to buy new items, repair unserviceables (in-house or contractual), redistribute stock, make direct shipments to customers from vendors and other supply decisions which are possible in the real system. The scenario is a chronological sequence of events that occur in the supply system being managed by each group during the course of the simulation. Since the exercise is designed to simulate four years of operation, the scenario is divided into 48 time-frames, each representing one month of the four year span of the simulation.

Two of the major objectives of the simulation are to provide supply managers, in an atmosphere of real life complexity, opportunities to:

(1) Practice decision making in a situation where errors are not costly.

(2) Practice better supply management using the latest scientific management techniques. These include economic order quantities, statistical safety levels, probabilistic returns of repairables and different methods of forecasting demand.

In addition to the training benefits, the more sophisticated managers can utilize CALOGSIM as a laboratory to experiment with and measure the results of some of the more recent scientific and mathematical techniques applicable to logistics systems.

CAPERTSIM (A Computer Assisted PERT Simulation)

CAPERTSIM is based upon an application of the principles of PERT (Program Evaluation and Review Technique) which is recognized as one of the more powerful management tools to emerge into the forefront in recent years. This system is now almost universally utilized by the military services in managing their complex weapons systems projects. The Navy Department has stated that it was due primarily to PERT that they were able to have in being an operational [illegible]'s missile system two years ahead of schedule.

One may ask, "Just what is this new management tool and what makes it so powerful?" In concept one may reply, "It is simply a method of planning and controlling interdependent activities in order to determine potential problem areas and insure timely completion." However, in detail, PERT is much more than this. It is an entire system for planning and evaluating progress in the development of complex weapon systems. It is the scheduling and monitoring of thousands of items which make up a complex weapon system, many of them involving research and development which in itself is difficult to schedule and/or predict precisely, that makes weapon system management so difficult and the PERT technique so valuable.

An apparently unimportant event at a subsystem level, which could easily escape the attention of an overall weapon system manager, could mean a major system delay at some point in the future unless corrective measures are taken immediately.

For example, a two-week delay in the shipment of a tiny bearing for a gyroscope might seem unimportant to the system manager of an ICBM missile program. But this could result in a three-month delay in actual flight testing because of a shortage of launch facilities when the missile finally is delivered.

Using linear programming techniques adapted to include statistical (probability) concepts, and digital computers to speed the analysis of interrelationships between the thousands of weapon system elements, PERT provides managers with all information required to effectively manage the system.

Recognizing the critical need for training project managers in the use and application of this concept, USALMC, developed a PERT simulation involving cost-time trade-offs.

CAPERTSIM simulates the management of a complex developmental type project utilizing the management tool known as PERT. Activities and events within the project are depicted graphically in network form showing all interrelationships and time dependencies. Each activity has an associated cost as a function of time.

The function of the Simulated Project Management Group is to achieve timely project completion at minimum cost in light of statistical probabilities. CAPERTSIM is designed to cover six reporting periods, each of which simulates 20 days of real time. The students are furnished computer-prepared situation summaries at the end of each reporting period which clearly pin-point critical and potentially critical elements in the program and their probable effect on the overall system schedule. A scenario is programmed into the computer which simulates the passage of time and the effects of simulated real life events. These in turn evoke management decisions on the part of the students.

Some of the objectives of CAPERTSIM are to provide students, in an atmosphere of real life complexity, experience in:

1. Evaluating cost-time relationships and trade-offs.
2. Evaluating effects of alternative courses of action.
3. Decision making under uncertainty.
4. Group dynamics.
5. Utilizing PERT as a scientific management tool in conjunction with a computer to assist in implementing management by exception techniques.

CARESIM (A Computer Assisted Repair Simulation)

In this simulation managers are placed in a maintenance environment in which they must determine the optimum maintenance policy for replacement of parts on each of several items of equipment. It permits the testing of four different repair part replacement policies in order to determine the optimum policy. These policies are as follows:

1. Replace individual parts only as they fail.
2. Replace all parts when one fails.
3. Replace a part which fails and replace all other parts which have been in use for X no. of hours or more.
4. Replace all parts after X no. of hours of operation or at time one part fails, whichever is earlier.

Each piece of equipment is tested for a period of fifty thousand hours of operation under each of the different policies to determine what the cost would have been had that policy been implemented. The computer simulates the 50,000 hours of operation in approximately five seconds. This is done by the Monte Carlo technique, knowing the probability distribution of failure for each piece of equipment. In effect, students are able to measure what the results of a particular decision would be before they actually implement it.

As more advances are made in the state of computer technology and as more experience is gained in computer simulations, future uses of computer simulations as an aid to education and decision making should increase considerably. Complex operations will be reduced to mathematical models capable of being programmed on computers with complete quantification of all the interrelationships existing within the operation. Managers and students will then be able to more adequately measure the results of a particular decision with a very high degree of validity. The use of probabilistic rather than deterministic models will also greatly enhance the education and decision making process. As a result, significant advances in the scientific methodology applicable to decision making should be made and result in more efficient and effective decisions being made at every level of operation.

THROWING DICE BY COMPUTER

Alan L. Breitler
Associate Professor

Montgomery College
Takoma Park, Maryland 20012

In order to teach students at Montgomery Community College about probability and the way it manifests itself in our daily lives, a computer program DICE (VERSION 1) was devised which permits students to "throw" a pair of dice 5000 times in a few seconds, and then analyze the results.

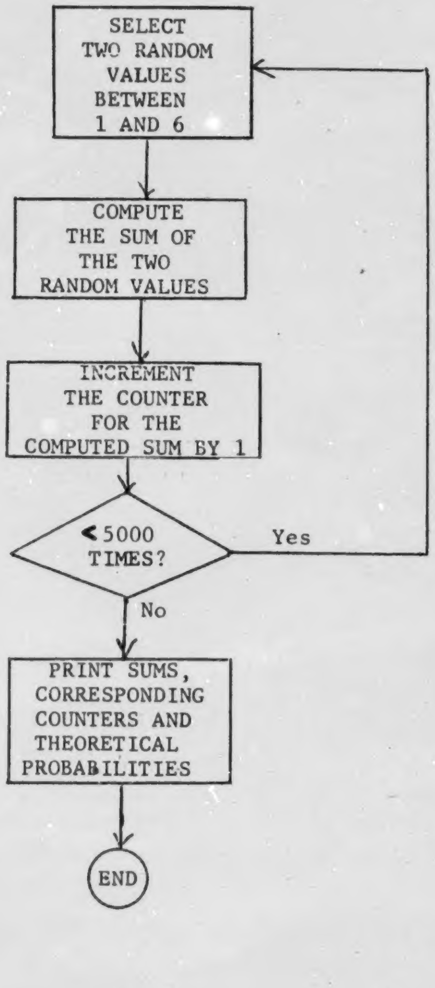
It can be seen that the distribution of probabilities is symmetric about 7, which has the highest probability of occurring.

To verify these theoretical probabilities the student runs the DICE 1 program which "throws" the dice 5000 times and prints actual probabilities. (The student is encouraged to use another program to get a chi-squared value on the actual versus theoretical probabilities to determine the significance of any difference that may appear).

Once the student is satified that the probabilities prediced by theoretical considerations are verified in practice he can proceed to the next step - simple loading.

The simple loading program DICE (VERSION 2) prohibits certain values from appearing on each of the two dice, thus one die may be prohibited from coming up with a 3 and the other die will never appear with a 4. Prohibiting this combination, many students believe, will somehow decrease the number of 7 totals and the probability of a 7 total appearing. Students are generally surprised when they discover that a greater probability of 7 and more sevens result from such loading.

Program Flow Chart for DICE (VERSION 1)


SELECT TWO RANDOM VALUES BETWEEN 1 AND 6
COMPUTE THE SUM OF THE TWO RANDOM VALUES
INCREMENT THE COUNTER FOR THE COMPUTED SUM BY 1
<5000 TIMES?
Yes
No
PRINT SUMS, CORRESPONDING COUNTERS AND THEORETICAL PROBABILITIES
END

	1	2	4	5	6
1	2	3	5	6	7
2	3	4	6	7	8
3	4	5	7	8	9
5	6	7	9	10	11
6	7	8	10	11	12

The above chart indicates what happens when 3 is barred on one die and 4 is barred on the other.

Theory indicates that the frequency with which a particular total of 2 dice will appear is based on the following table:

		Die #1					
		1	2	3	4	5	6
Die #2	1	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

The probabilities of each total are calculable by means of the following:

$$P = \frac{\text{\# of times total appears in table}}{\text{total \# of possibilities}}$$

The total number of possibilities is 36 (i.e., 6 x 6). The probabilities and approximate percentages for each total are:

Total	P	%	Total	P	%
2	1/36 =	.0278	7	1/6 =	.1667
3	1/18 =	.0556	8	5/36 =	.1389
4	1/12 =	.0833	9	1/9 =	.1111
5	1/9 =	.1111	10	1/12 =	.0833
6	5/36 =	.1389	11	1/18 =	.0556
			12	1/36 =	.0278

Probabilities now are as follows:

Total	P	%	Total	P	%
2	1/25 =	.04	7	1/5 =	.2
3	2/25 =	.08	8	3/25 =	.12
4	2/25 =	.08	9	2/25 =	.08
5	2/25 =	.08	10	2/25 =	.08
6	3/25 =	.12	11	2/25 =	.04
			12	1/25 =	.04

The probability of a 7 total has increased from 1/6 to 1/5!

Many other interesting combinations have been tried with sometimes equally surprising results.

Some students, unhappy with the arbitrary nature of the simple loading, decided to write a program which more accurately simulated real events. They constructed a routine which permits fractional loading in which a certain percentage of values are rejected.

Thus a 3 on one die might be acceptable 60% of the time and a 4 on the other 45% of the time. The implications of the results are not yet clear, but these students are gaining a greater knowledge of probability theory through their computer - aided investigation.

```
READY


LIST


DICE           15:28           04-DEC-74


1 PRINT"VERSION 1 OR 2";
2 INPUT V
3 IF V=2 THEN 180
5 DIM C(13)
10 RANDOMIZE
20 FOR X=1 TO 5000
30 D1=INT(RND*6+1)
40 D2=INT(RND*6+1)
45 IF V=2 THEN 210
50 T=D1+D2
60 C(T)=C(T)+1
70 NEXT X
80 PRINT"TOTAL","COUNT","  %  "
100 FOR Y=2 TO 12
110 PRINT Y,C(Y),.01*INT(C(Y)/.5)
160 NEXT Y
170 GOTO500
180 PRINT"ENTER FORBIDDEN VALUES";
190 INPUT F1,F2
200 GOTO10
210 IF D1=F1 THEN 30
220 IF D2=F2 THEN 40
230 GOTO50
500 END

READY
RUN

DICE           15:30           04-DEC-74


VERSION 1 OR 2 ?1
TOTAL          COUNT          %
 2              139           2.78
 3              262           5.24
 4              463           9.26
 5              593           11.86
 6              636           12.72
 7              788           15.76
 8              715           14.3
 9              572           11.44
 10             416           8.32
 11             277           5.54
 12             139           2.78


TIME:  5.35 SECS.
```

```
RUN

DICE           15:33           04-DEC-74


VERSION 1 OR 2 ?2
ENTER FORBIDDEN VALUES ?1,6
TOTAL          COUNT          %
 2              0             0
 3              205           4.1
 4              428           8.56
 5              630           12.6
 6              768           15.36
 7              991           19.82
 8              797           15.94
 9              589           11.78
 10             391           7.82
 11             201           4.02
 12             0             0


TIME:  6.60 SECS.

READY
RUN

DICE           15:31           04-DEC-74


VERSION 1 OR 2 ?2
ENTER FORBIDDEN VALUES ?3,4
TOTAL          COUNT          %
 2              208           4.16
 3              397           7.94
 4              414           8.28
 5              403           8.06
 6              571           11.42
 7              964           19.28
 8              616           12.32
 9              424           8.48
 10             392           7.84
 11             432           8.64
 12             179           3.58


TIME:  6.91 SECS.

READY
```

ADMINISTRATIVE AND INSTRUCTIONAL COMPUTING IN MINORITY INSTITUTIONS

John W. Hamblen
University of Missouri-Rolla
Gerald L. Engel
Virginia Institute of Marine Sciences
Vincent H. Swoyer
University of Rochester
Lawrence Oliver
National Science Foundation

Background

Many of the minority institutions are feeling the pressure to obtain computer facilities for instructional purposes. Because they are relative small (enrollments at most are less than 5,000), the immediate question which arises is "can we also make use of the computer for administrative, functions, such as registration grade reporting, accounting and alumni affairs," and, of course, the answer is "That depends--."

During the late fifties there were numerous opportunities for institutions (even large universities) just getting started to attend meetings and workshops to learn, share and discuss various alternatives, approaches, and techniques of using the computer for research, instruction and administrative data processing. This continued through the sixties for smaller institutions.

Administrative Computing

Such meetings as the Machine Records Conferences, the annual meetings of the Association for Educational Data Systems, CAUSE, COMMON, CUETUG, etc., became more sophisticated and in some cases equipment dependent (user groups). Many of the minority institutions have neither the equipment resources or experiences to participate in such meetings today. On the other hand a few have such resources and experiences which can and should be shared with their sister institutions. It has been proposed that the Association for Educational Data Systems be the catalyst for this exchange.

Conferences such as the Machine Records Conference and the annual meetings of AEDS attract primarily administrators from institutions having large and sophisticated equipment utilizing in most cases advanced computing technologies and concepts well beyond the experience and resource capabilities of most minority institutions.

The objective of the proposed workshops is to provide stimulation and assistance to four hundred college administrators from minority institutions. The most likely offices to be represented are registrar, admissions, accounting, alumni, development and purchasing. The proposer feels that the introduction of the administrators to computers and their uses in administration would greatly strengthen their ability to consider alternatives in computer selection and participate in information system analysis and design.

Instructional Computing

During the past six years, the National Science Foundation has partially supported six annual conferences on Computers in Undergraduate Curricula (Iowa, 1970; Dartmouth, 1971; SREB, 1972; Claremont, 1973; Washington State, 1974; and Texas Christian, 1975).

From the start, the Conferences on Computers in Undergraduate Curricula attracted primarily faculty from institutions having large and sophisticated equipment utilizing in most cases advanced computing technologies and concepts well beyond the experience and resource capabilities of most minority institutions.

In 1974 Lincoln University of Missouri received a $151,700 grant from the National Science Foundation's Minority Institution Science Improvement Program to provide for the First Conference on Educational Computing in Minority* Institutions (ECMI/1). The planning and conduct of the conference was contracted to the Computer Science Department of the University of Missouri-Rolla (UMR). Hamblen served as Conference Coordinator and was assisted by Joseph W. Trigg, Instructor of Mathematics, Lincoln University. Basic policy guidelines were provided by a Steering Committee of representatives from nine minority institutions.

---

*"Minority" for purposes of the grant has been defined as "four-year traditionally black."

The members of the Steering Committee and their respective institutions are: Dr. James Frank, Chairman of the Committee and President of Lincoln University; Dr. James Kinard, Benedict College; Dr. Jesse C. Lewis, Jackson State University; Sister Patricia Marshall, Xavier University of Louisiana; Mr. Jesse J. Mayes, Federal City College; Dr. Albert Miller, Delaware State College; Mr. Grover Simmons, Atlanta University; Mr. Fred Stone, Tuskegee Institute; and Dr. Roger K. Williams, Morgan State College.

The Conference was held in the Sheraton-Biltmore Hotel of Atlanta, Georgia during March 23-27, 1975. It was directed at acquainting faculty of minority institutions on the use of the computer to assist them with the teaching and/or learning process, and to inform them of hardware and instructional techniques that may be employed in improving their science education programs.

Approximately one hundred, four-year, traditionally black institutions were invited to submit at least three applications. Two hundred participants were selected from the eighty-six institutions which responded. Each of these eighty-six institutions were represented by at least one faculty member. Selection was based on priorities assigned to the applicants by their institutions. A high priority was given to an applicant who could influence the promotion and/or development of computing at his/her institution, and to those who did not have extensive background in computing.

The major areas represented at the Conference were the mathematical, physical, natural, social and management sciences. These areas were represented by twelve disciplines. Each participant attended five group sessions with 5-15 faculty of the same discipline. At these sessions, faculty who had used the computer in their classes presented their materials to the attendees from their discipline. The group leader demonstrated what he has done and how he used the computer in his classes. In addition the four-day conference consisted of five general sessions and seven sessions of a programming short course in a language of the participants choice from FORTRAN, BASIC, and APL.

ACM Consulting Service [ENGEL]

History

The ACM has always regarded educational programs as one of its major missions. Educational activities within ACM led to the publication of curriculum recommendations in computer science in March, 1968, known internationally as "Curriculum '68" [17]. The publication of this document along with the report of the COSINE Committee of the Commission on Education of the National Academy of Engineering [9], and the Park City Conference on Computers in Undergraduate Education [29], all within a year went a long way to define the discipline of Computer Science and focus on key issues in computer uses in education.

The Curriculum Committee on Computer Science of ACM, who prepared "Curriculum '68", believed that it was necessary for the document to be interpreted to potential users. The initial vehicle for this was the ACM Visiting Scientist Program, supported by the National Science Foundation, which arranged for speakers to come to college campuses. In reviewing the accomplishments of this program, it was determined that a broader program was needed than one in which a computer scientist spent part of a day giving a lecture on computer science education.

Thus in June, 1969, a proposal was submitted to the National Science Foundation to provide funds for a more in-depth consulting visit to college campuses in which consultation would be offered in four basic areas:

1) Computer Science Curricula - ranging from the course content of an introductory course in computer science to a complete undergraduate curriculum.

2) Computer Facilities - emphasizing the alternatives available to the schools such as networks, batch systems, time-sharing systems, and commmercial services.

3) Computer Uses in Education - showing the activities that are going on in various subject areas where computers are being used and supplying information on where materials could be obtained.

4) Administrative and Budget Matters - dealing with questions such as where, in the school, computer science courses should be offered, who should control the computing facilities, how should faculty interested in computing be trained and recruited, and where are funds available to support computing on campus.[1]

This proposal was funded in October, 1969 by the National Science Foundation, and from that time until July, 1973, the College Consulting Service functioned as a service of ACM.

---

[1] From the proposal

Procedures and Activities

The mechanics for the consulting visits were constructed to be quite flexible for the college visited:

> Visits are usually for two days, although travel requirements may cut into part of this time.
>
> The agenda for the visit is the responsibility of the school. It should be sent to the consultant before the visit. It may include a formal talk to students and faculty, but small group meetings and other informal activities have proven most effective. There should be some opportunity to visit with a few interested students.
>
> Topics such as various aspects of computer use, budget, problems, facilities and equipment, curriculum, faculty education, and administration can best be covered by scheduled meetings with small groups of appropriate faculty and administrators.
>
> A private talk toward the end of the visit with the president or senior academic dean is of considerable value.[2]

Notices of the existence of the program were sent by personal letter to the presidents of most four-year institutions in the country. In addition, announcements of the program were made in the trade literature and at selected professional meetings. Applications were submitted to the program director who made the selection of institutions to be visited and assigned the consultant. Eligibility was left somewhat open.

> The program is intended for undergraduate colleges and small universities where undergraduate education is the major objective. Institutions granting the bachelor's degree and accredited by a regional accrediting agency may apply.[3]

In the life of the program approximately 100 trips were made.

The distinction of "small' became quite difficult to make, in that in some cases state colleges with quite large enrollments were just getting started in computer work, and needed as much assistance as the under 2,000 student liberal arts college. Toward the end of the program, several two-year colleges were visited to assess the nature of the problems of computing at these institutions.

The relative emphasis over the years of the lifetime of the Consulting Service showed some changes. Perhaps the most dramatic change was from relatively little discussion of computer science curriculum early in the program to a point where it was the main purpose of the visits near the end of the program.

The early visits dealt mainly with questions of equipment and staff. Interestingly, the institutions in their applications generally stressed equipment selection as their main problem, and the visitor generally found the main problem to be staff.

In the early visits, few schools were even considering a computer science major. Such programs were considered only appropriate for large universities and engineering schools. Furthermore, staff for such programs was not available. With time the staffing problem became somewhat less difficult as graduate computer science programs produced more individuals with advanced degrees, and the schools found that the demand for computer science from entering students and the market place made it most desirable to introduce such programs.

Recommendations

Since the visits were not all alike, there can not be a single set of recommendations that apply to all schools. There were certain trends, however, that were clearly seen in reviewing the series of consultant reports.

Perhaps the single most common problem was the competition, either actual or potential, between administrative data processing and academic computing. Many schools had justified acquisition of their equipment on the savings it would give in administrative record keeping, and even where this was not the case, many administrators looked at the expensive array of equipment in the computer center, and were considering how it could be used for administrative work. The unfortunate part of this is that administrative data processing often became the "tail wagging the dog", and with little, if any cost savings. More information was generally produced by the computerization of administrative activities, but there was seldom any reduction in cost. What usually did happen, however, was that the personnel assigned to the computer center were required to devote a great deal of their time to the implementation of administrative programs, and often much of the available computer time was devoted to this purpose.

---

2 from the brochure describing the ACM College Consulting Service

3 *Ibid*.

In those institutions with enrollments of under 1,500 students, Hamblen suggested in the Proceedings of the Rand Conference [23] that there is a question as to whether any administrative computing should be done. He goes on to observe that for schools of under 2,500 students, it is difficult to justify a separate machine for administrative use.

Noting these observations, it does appear that the approach of separate machines for academic and administrative computing is the most desirable thing to do. In this way, conflicts over use should not arise. In many cases this is impractical, and there are some other possibilities. Most schools operate in such a way that students have little opportunity to use the computing equipment in the morning hours, while having needs for computer time in the afternoons and evenings. This could then suggest a firm scheduling scheme which would allow both groups uninterrupted time. Such a schedule should be under the control of the computer center director who would insure fair treatment of everyone.

As important as the problem of scheduling machine time is for administrative and academic work, staff scheduling may be more critical. All too often schools have been in a position where all of the computer staff time is devoted to the development and maintenance of administrative computing. In this case, it appears desirable to assign a separate group within the center, reportable to the director, to handle administrative computing. In this way, there will be a minimum of competition, and the true cost of the administrative operation will be determined.

The position of the computer center in the institutional organization is a recurring source of discussion. Here again, almost every possible combination from part of an academic department to an independent agency working under contract to the school was seen. It is clear that local circumstances come into play here, but as a general rule for those centers serving all aspects of the campus, the director of the center should be relatively independent, and report to an administrative officer above those he serves. This generally would imply a line to the president of the college or a senior academic vice president. In this way, the center should not become dominated by any single user.

In the life of the Consulting Service it was determined that the recommendation of specific hardware was impossible. As a general rule of thumb, it was determined that a school should anticipate a minimum expense of $50,000 per year to run a computer center and this would be approximately equally divided between equipment and staff. As the center grows this 50-50 split is likely to remain and it was found that the institutions were less likely to anticipate the staff costs than the equipment cost.

The other common recommendation regarding equipment involved the careful analysis of the uses of the system. In particular it is often the case that the equipment necessary for administrative data processing is quite different from that needed for academic computing. This becomes especially important when dealing with small machines, and again points out the importance of justifying and specifying equipment on what its use will be.

The problems of finding adequate staff seems to have improved in recent years. Computer science students, though still not plentiful, are certainly more available, and individuals with experience and backgrounds in the applications of computing are also available. A factor that is working against this, however, is that financial problems have caused many institutions to put a freeze on new appointments, and there has been a tendency to reassign people into the computer area rather than recruiting individuals trained in this field.

The position of director of the computer center calls for coordinating and developing all aspects of computing on campus. It also involves a key role in the selection of equipment, staff and services. There are no specific training programs for such individuals, though some courses in management of computer centers are appearing in university curricula. Where possible the person selected for this position should have experience in management of a computer installation, and in implications of computer applications. This position should be distinct from a position in teaching computer science, or any other department since the center should be independent of other organizational units of the campus; a courtesy faculty appointment, however, and occasional teaching is not inappropriate. There is little practical reason for the computer center director to hold a doctorate, though, it must be recognized that this individual will have a great deal of interaction with the faculty, and it may be necessary for him to hold this credential to achieve the proper respect of the faculty.

The rest of the staff of the center is closely tied to the nature of the work, and cannot be generally specified. It should, however, be recognized that if an organization wants quality programming and operations, it is necessary to have a staff of professionals, and not operate as many schools do, by leaving the problems of program development and implementation to part time help and amateurs.

Computer Science educational programs should be staffed separately from the computer center, and indeed the computer science department or group should be regarded as just another user of the computer facility. In the case of the academic program it appears critical that the positions be filled with computer scientists. The field has developed to the position of a field of study in its own right, and it is difficult to see how someone not primarily engaged in the field can keep up with the developments necessary to teach at the college level. How large a staff in computer science is necessary depends on the level of activity, but it is difficult to see how there can be less than one full-time person in this capacity.

The necessary ingredient to the development of innovative uses of computers in education appears to be administrative support and encouragement. This can come in many forms and in some ways the easiest is in the recruitment of new faculty members. Many students completing their work are actively engaged in the uses of computing, and are anxious to continue this in their professional life. Another approach is to institute an active program of bringing visitors to the campus who are involved in computer uses in education. In a two or three day period, such a visitor can present several lectures, and at the same time push the faculty into such activities on their own.

Other activities that have proved successful have been specially funded training and orientation programs for the faculty, and support of innovative educational computer uses through summer developmental grants. In all such cases it is necessary that the administration of the school be committed to such activities and insure that the faculty member so involved receives the proper academic credit for what is a time consuming and demanding task. It should be noted that in many ways the smaller schools, without the emphasis on "classical publication", have a better chance to have faculty members devote their time to innovative computer use.

In considering instruction in computer science the first point to observe is that such instruction has a rightful place in a college of arts and sciences. Many arguments have been presented on this point and it will serve little purpose to repeat them here. Granted that such instruction has a place in the curriculum the question then is how much how soon. Here again local conditions affect the decision, but nearly all schools with active computing activities have some form of computer science instruction. This is what keeps the students excited and active in computing, and in fact it is often the students that insist that such instruction be offered.

In all too many cases, however, schools have looked at computer science and determined that this is an active field, offering job potential for the students, and perhaps even serving to attract students at a time when many colleges are suffering from declining enrollments. They will often then select a computer user on campus and put him in charge of a "program" that does little more than offer several courses in different programming languages. The fact that computer science is a vital active field that should be represented at most colleges does not excuse such a procedure. It must be recognized that to institute a computer science program is both expensive and time consuming, and requires a commitment by the school at a time that limited resources may demand cutting back in some other areas.

LaFrance and Roth in reporting on a Workshop on Computer Science for Liberal Arts Colleges [22] give some indications as to how the development of a department should proceed. Recognizing that few liberal arts schools would be able to introduce a major, they called first for a strong development of a first course similar to course B1 of "Curriculum '68" [17] or Course 1 of Austing and Engel's report [4], and then as interest and staff develop, offer courses involving things like machine organization and programming, information structures and a survey of computer applications; courses 3, 4 and 2 respectively of Austing and Engel [4]. At such a time as these courses are under control, further work could be added as faculty and student interest dictates. Offering this kind of program would provide a firm base of knowledgeable students, would supply necessary service courses for a variety of students, and combined with an adequate selection of courses from other departments would meet minimum entrance requirements for admissions to graduate programs in computer science for interested students, or for direct entrance into the computing profession.

It should be noted that offering this much of a program requires a full time faculty member, and clearly, to expand these offerings would require more. It is also clear that such courses should be offered as computer science and not in some other department. To serve the service need of the program, a selection of one or more of these courses could be taken with no other prerequisites, and, to offer them in a mathematics department, for example, would only serve to drive away students who are not mathematically oriented.

Such a program, then is within the reach of any school willing to commit itself to one staff member in computer science, it will serve to form a basis for future development of a more advanced program, it will serve a multi-track service need, and it will offer enough background for further work in computing.

The library resource is often neglected in considerations of the development of computer science and computer uses in education. Often the expenses of obtaining equipment are so great that little time is given to think out this pressing need. Most of the references at the end of this paper contain extensive bibliographies of materials appropriate to computer science education and computer uses in education. The selection of such items would provide a start at a good working collection.

It is often easy in addressing questions of educational computing, to overlook the obvious in the specification of details. The obvious needed ingredient to allow for excellence of either computer science instruction, or computer uses in education, is free and easy access to the computing equipment, preferably utilizing one of the user oriented systems that provide fast turnaround and clear diagnostics. This must be the first order of business.

Sources of Information

One of the major functions of the consultants in this program, was to direct interested people at the institutions visited to sources of additional information. No listing of such materials could be complete, however, a few items will be noted which have proved valuable.

Certainly any reading into educational computing should begin with the Pierce Committee Report [24]. Though, by and large, these recommendations for the proper level of support of educational computing have not been followed, they still serve as an excellent goal. In terms of additional documents in the full range of computing and computer services, three conference proceedings, the World Conference on Computer Education, 1970 [25], the Rand Conference, 1970 [23], and the Park City Conference, 1968 [29] are excellent. Computer Surveys,[4] published by ACM provides a good introduction to various aspects of computing and should be in most libraries. Computing Reviews also published by ACM provides up to date information on available books, articles, and other materials.

ACM has a Special Interest Group on University Computer Centers (SIGUCC). This group publishes a quarterly newsletter dealing with various aspects of computer center administration. In June, 1972 SIGUCC sponsored a symposium on the administration and management of small college computer center, and proceedings are available which cover many of the problems associated with small centers [6].

A number of sessions have been held dealing with the uses of computers in various disciplines. Among these the Iowa Conference [30], the Dartmouth Conference [21], the Atlanta Conference [19], the Claremont Conference [32], and the IIT Conference [5] provide sources of valuable information and programs for computer applications. Sedelow [26] has prepared an outstanding survey on computer uses in the humanities. The Human Resources Research Organization (HumRRO) is engaged in an overall study of computer-based curriculum and has prepared a preliminary report [27].

In the area of applications, ACM has several Special Interest Groups dealing with computer uses in education (SIGCUE), social and behavioral sciences (SIGSOC), language analysis and studies in the humanities (SIGLASH), and computers and society (SIGCAS). All of these groups publish periodic bulletins and sponsor occasional meetings, usually with proceedings available.

To begin reading on computer science education "Curriculum '68" [17] is a good start. Work reported by LaFrance and Roth [22] and Austing and Engel [4] have attempted to put computer science programs into the perspective of the small colleges. Reports by Wegner [31] and the Committee on the Undergraduate Program in Mathematics [7] present some additional ideas regarding computer science programs. The ACM Curriculum Committee on Computer Education for Management [3, 28] presents outlines of work in information systems. The COSINE Committee of the Commission on Education National Academy of Engineering specifies the role of computer science in electrical engineering [9], and in subsequent reports outlines specific courses; computer organization [11], a computer-oriented first course in electrical engineering [10], digital subsystems [12], digital systems laboratory course [15], and operating systems principles [14].

ACM has a Special Interest Group on Computer Science Education (SIGCSE) which publishes a quarterly bulletin, and this group has held three technical symposia for which proceedings are available [1, 2, 18].

[4] Information regarding this journal and other ACM activities, such as the Special Interest Groups, may be obtained by writing ACM Headquarters, 1133 Avenue of the Americas, New York, New York 10036.

Future

ACM is currently seeking support for a similar consulting program for minority institutions.

References

[1] Robert M. Aiken (ed.), Papers Presented at the ACM SIGCSE Technical Symposium on Academic Education in Computing Science, November 16, 1970, Houston, Texas, Association for Computing Machinery, New York, 1970.

[2] ______________, Papers Presented at the ACM SIGCSE Second Symposium on Education in Computer Science, St. Louis, Missouri, March, 1972, Association for Computing Machinery, New York, 1972.

[3] R.L. Ashenhurst (ed.), Curriculm Recommendations for Graduate Professional Programs in Information Systems, A Report of the ACM Curriculum Committee on Computer Education for Management, Communication of the ACM, Vol. 15, No. 5, May, 1972, 364-398.

[4] Richard H. Austing and Gerald L. Engel, A Computer Science Course Program for Small Colleges, Communications of the ACM, Vol. 16, No. 3, March, 1973, 139-147.

[5] Ronald Blum (ed.), Computers in Undergraduate Science Education, Conference Proceedings, Chicago, Illinois, August 17-21, 1970, Commission on College Physics, College Park, Maryland, 1971.

[6] Harris Burns, Gerald L. Engel, and John H. Esbin (eds.), The Administration and Management of Small College Computer Centers, Proceedings of a SIGUCC Symposium June 9-10, 1972, Atlanta, Georgia, University Computer Center, Iowa City, Iowa, 1973.

[7] Committee on the Undergraduate Program in Mathematics (CUPM), Recommendations for an Undergraduate Program in Computational Mathematics, CUPM, Berkeley, California May, 1971.

[8] Conference Board of the Mathematical Sciences, Recommendations Regarding Computers in High School Education, Conference Board of the Mathematical Sciences, Washington, D.C., 1972.

[9] COSINE Committee, Computer Science in Electrical Engineering, National Academy of Engineering, Washington, D.C., September, 1967.

[10] ______________, Some Specifications for a Computer-Oriented First Course in Electrical Engineering, National Academy of Engineering, Washington, D.C., September, 1968.

[11] ______________, An Undergraduate Electrical Engineering Course on Computer Organization, National Academy of Engineering, Washington, D.C., October, 1968.

[12] ______________, Some Specifications for an Undergraduate Course in Digital Subsystems, National Academy of Engineering, Washington, D.C., November, 1968.

[13] ______________, Impact of Computers on Electrical Engineering Education - A View from Industry, National Academy of Engineering, Washington, D.C., September, 1969.

[14] ______________, An Undergraduate Course on Operating Systems Principles, National Academy of Engineering, Washington, D.C., June, 1971.

[15] ______________, Digital Systems Laboratory Courses and Laboratory Developments, National Academy of Engineering, Washington, D.C., March, 1971.

[16] ______________, Minicomputers in the Digital Laboratory Program, National Academy of Engineering, Washington, D.C., April, 1972.

[17] Curriculum Committee on Computer Science, "Curriculum '68, Recommendations for Academic Programs in Computer Science", Communications of the ACM, Vol. 11, No. 3, March, 1968, 152-197.

[18] Gerald L. Engel and Robert M. Aiken (eds.), The Papers of the ACM SIGCSE Third Technical Symposium on Computer Science Education, Columbus, Ohio, February 22-23, 1973, Association for Computing Machinery, New York, 1973.

[19] John W. Hamblen (ed.), Proceedings of the 1972 Conference on Computers in the Undergraduate Curricula, June 12-14, 1972, Atlanta, Georgia, Southern Regional Education Board, Atlanta, Georgia, 1972.

[20] IFIP Technical Committee for Education, Computing Education in Secondary Schools, an Outline Guide, AFIPS, Montvale, New Jersey, September, 1971.

[21] Thomas E. Kurtz (ed.), Proceedings of the Second Annual Conference on Computers in the Undergraduate Curricula, Dartmouth College, June 23-25, 1971, The University Press of New England, Hanover, New Hampshire, 1971.

[22] Jacques LaFrance and R. Waldo Roth Computer Science for Liberal Arts Colleges, A Report of a Workshop held at Wheaton College, Wheaton, Illinois, July 12-24, 1972, SIGCSE Bulletin, Vol. 5, No. 1, February, 1973, 70-76.

[23] R.E. Levien (ed.), Computers in Instruction, their future for Higher Education, Proceedings of a Conference held in October, 1970, Los Angeles, California, The Rand Corporation, Santa Monica, California, July, 1971.

[24] President's Science Advisory Committee, Computers in Higher Education, U.S. Government Printing Office, Washington, D.C., February, 1967.

[25] B. Scheepmaker and K.L. Zinn (eds.), World Conference on Computer Education 1970, Papers of the First International Federation of Information Processing World Conference on Computer Education, August 24-28, 1970, Amsterdam, The Netherlands, Science Associates/International, New York, 1970.

[26] Sally Yeates Sedelow, The Computer in the Humanities and Fine Arts, Computing Surveys, Vol. 2, No. 2, June, 1970, 89-110.

[27] Robert J. Seidel, Beverly Hunter and Martin L. Rubin, A Study of Computer-Based Curricula: The Process of Change, Human Resources Research Organization, Alexandria, Virginia, November, 1971.

[28] Daniel Teichroew (ed.), Education Related to the Use of Computers in Organizations, Communications of the ACM, Vol. 14, No. 9, September, 1971, 573-588.

[29] William Viavant (ed.), Proceedings of the Park City Conference on Computers in Undergraduate Education, Park City, Utah, September 8-13, 1968, University of Utah, Salt Lake City, Utah, 1969.

[30] Gerard Weeg (ed.), Proceedings of a Conference on Computers in the Undergraduate Curricula, Iowa City, Iowa, June 16-18, 1970, University Computer Center, Iowa City, 1970.

[31] Peter Wegner, A View of Computer Science Education, The American Mathematical Monthly, Vol. 79, No. 2, February, 1971, 168-179.

[32] Fred Weingarten (ed.), Proceedings of a Fourth Conference on Computers in the Undergraduate Curricula, Claremont, California, June 18-20, 1973, The Claremont Colleges, Claremont, California, 1973.

Minicomputers [SWOYER]

[An experiment was funded by NSF in August, 1970, which assisted ten colleges' acquisition of mini-computers with coordinating supervision by the Southern Regional Education Board Computer Sciences Project.]

The ten colleges in the experiment exhibited a variety of computing needs and a diversity of preference in choice of computer vendor, configuration, and mode of operation. At the time of the start of the experiment, circa 1969, the purchase of a mini-computer by a small college posed an element of risk few felt they could afford. There was almost no experience based on which to rate the expected performance of a mini-computer in an educational environment, and although many faculty had previous experience in the use of computers, almost none were well versed in the elements of systems selection, software, hardware, or operational considerations. The grants from NSF were intended to partially off-set the risk, and the introduction of the SREB office of the Computer Science Project, plus support for a series of follow-on consultant visits, assisted with the establishment of reasonable initial computer operations on each campus. The experiment required institutional commitment to provide information about progress and use of the facilities through the three year project period.

The equipment selection process, and associated plans for operation, occurred in a tentative form in February, 1970. In many cases, revisions to original plans continued right up to a month or so prior to delivery. For most institutions, this was the first experience ordering a computer. The early 1970 time frame deserves some reflection. Most major mini-computer vendors were in the process of releasing new lines of equipment which are now considered commonplace. Data General had barely entered the market with its first Nova. Digital Equipment Corporation had just delivered the first of its PDP-11 series. Hewlett-Packard was just beginning to extend the popular HP2000 series computers. It was a time when much of the minicomputer equipment was new even to the sales representatives. To the uninitiated consumer, there was genuine cause for bewilderment. Several colleges planned a system in March of 1970, and by the time of the experiment's start in August discovered new equipment had come into the market with greater appeal. The case for Data General equipment, for example, found some attractively priced new minicomputers available which were completely unknown when the initial systems were evaluated five months earlier.

Some General Observations about Equipment Selection.

Although the ten colleges varied somewhat in the objectives to be satisfied by the minicomputer system, some initial generalizations were apparent.

1. Institutions primarily interested in economical use by fairly large numbers

of students opted for a batch mode of operation.

2. Three "least costly" systems, all batch or single-user oriented, were priced around $20,000 for a complete system. No single vendor was at the low-cost end of the scale, with a Data General Nova, a Digital Equipment PDP-8, and a Hewlett-Packard 2114 each contending for lowest price honors.

3. Irrespective of vendor, most institutions opted for a high-speed paper tape reader/punch.

4. Five institutions chose to operate multiple terminals in a time-shared mode. Of these, all but one had at least 12,000 words of memory. (The only college attempting time-sharing with 8,000 words of memory found it needed more during the experiment).

5. The common characteristic separating the "low-cost" $20,000 systems from the others (ranging upward from $40,000) was the exclusion or inclusion of a high-speed rotating secondary memory device (either a disk drive or a drum).

6. All time-shared systems had a disk or a drum.

7. The two "highest-priced" systems ($91-93,000) were both manufactured by Hewlett-Packard. One has batch oriented and the other time-shared.

8. Although equipment from General Electric, Wang Laboratories, Xerox and Honeywell was included in the evaluation, the "big three" minicomputer vendors (DEC, Data General, and Hewlett-Packard) ended up as the only manufacturers represented.

9. Every installation planned to use BASIC, nine as the primary language. About half felt at least some FORTRAN was important.

Some Conclusions and Suggestions

The major findings of the experiments are presented here with an aim to assist other institutions considering a minicomputer for instructional use. A basic conclusion, expressed unanimously by the ten somewhat varied types of undergraduate institutions, is that a mincomputer system is the best buy source of computing power for most instructional activities. The combination of low cost, constant availability, control, and the many advantages of an on-campus facility make the minicomputer a preferred choice over known off-campus alternatives.

Many conclusions have a financial basis. Cost statistics and equipment configurations are summarized here. Table I of [SWOYER] shows the minicomputer system purchased by each institution, its purchase price (in 1970, for most equipment), and average annual costs. The average annual costs are broken down as "equipment" "salaries and benefits," and "other" components, and include calculation of annual costs per enrolled student. The latter figure corresponds with a "rule of thumb" statistic calculated in a number of previous experiments and projects. It is useful as a rough guide. In this instance, it must be noted that the annual costs per student include all costs, not simply equipment costs. The figure ranges from $5.40 to $33.24 per student per year, of which equipment costs (often the sole basis for calculating cost per student) range from $2.80 to $17.10 per student per year. In all, it may be noted that 12,450 full-time students were enrolled collectively by the ten institutions, with a mean total annual cost per student of $17.52. (Of that total, $9.30 is the mean annual cost per student for equipment.) The institution which had prior experience using one or two interactive terminals to a remote time-sharing computer, felt the minicomputer source was unquestionably superior. In addition to cost savings (institutions here, and in previous NSF experiments had experienced average annual costs of about $19 per student), greater availability, and other obvious advantages cited above, there was an expressed academic advantage. This is hard to quantify, but includes the advantage of having a facility on which software changes, operating system experiments, and other developmental activities can take place which would not be possible on a large or non-institutional facility. In addition, the on-site system created an atmosphere or center of activity supportive of creative thinking about computing.

Prior to the experiment, the colleges established rough budget guidelines for the various levels of minicomputer systems. The mean estimated total annual cost was $28,130. The actual mean total annual cost turned out to be $21,810. The average annual cost per student was estimated in advance to be $21.42, which compared to the actual cost of $17.52. Thus is was observed that costs were actually somewhat lower over three years than projected.

Equipment costs were found to average 53% of the total expenses. This component of the costs ranged from $4,700 to $22,000 per year with a mean of $11,580. It would be expected that costs for comparable systems would be somewhat below these figures today, although one must not hastily apply a factor of, say, 50% (which may accurately reflect the comparable price of a processor unit today versus

1970) too broadly. Processors and memories have come down in cost drastically. However, most mechanical devices (such as disk drives, tape drives, card readers) have experienced less reduction. Maintenance costs, included here as equipment expense, have increased. Overall, the minicomputer system of today should probably cost about 80% of the comparable 1970 version.

Other costs, however, have increased. Salaries, benefits, publications, and travel expenses generally total about 25% more than in 1970. Supply costs are 100% higher for paper items than just two years ago. Since salaries and benefits constitute 27% of total annual costs, and "other" costs (largely supplies and institutional overhead) represent the remaining 20%, the end result of a new 1974-75 operation should have a price tag close to par with 1970 costs.

The institutions were queried, after the fact, about typical questions, which might be asked by prospective institutional minicomputer users, and what the answers to those questions should be. The following are some typical questions, with responses, which could be asked by institutions exploring minicomputers.

Question 1: What is the most common oversight when planning a college minicomputer operation?

Response: Underestimating the amount of time needed by the person (faculty member) responsible for the operation, especially during the first year. A minimum of 50% of full-time should be allocated, with more time available during the first year, if possible.

Question 2: What is the primary equipment difference that most colleges would opt for if they could redesign the system after several years experience?

Response: The most common change would be a computer with a larger memory. (Eight of the ten institutions gave this response.)

Question 3: What vendor would you choose if you were to "do it over?

Response: Curiously, the response to this question, in 1973, was a preference for the same vendor as previously used. Some qualifications were attached, indicating that perfect satisfaction was not always present.

Question 4: How many students may be adequately supported by one interactive BASIC terminal?

Response: (Answered by the five institutions with interactive minicomputers.) Two institutions responded with 22 to 25, with three institutions answering 10 to 15. The variability reflected the amount of computing required of the "typical" courses. The same question has been addressed by other institutions with interactive experience. A consensus indicates 22 to 25 students per terminal to be a maximum number in a course with modest assignments. More institutions prefer a terminal to serve no more than 15 students in a computing course.

Question 5: Is there a high risk of equipment failure with a minicomputer? Is a maintenance contract a necessity?

Response: Component failure in equipment other than teletypes, card readers, disk drives, and tape drives are almost non-existent. Following an initial shake-down period (three to six months), every institution that shifted to "on call" service, in place of a monthly contract, reduced its maintenance costs. In most cases, the reduction was more than 80%. The consensus favored maintenance contracts only for mechanical components (if at all) with maintenance of teletypes separate from vendor-supplied service contracts in any event.

Question 6: Was there any indicator that would tend to identify in advance whether an institution would have a successful minicomputer operation?

Response: All institutions achieved successful operations, within a variety of objectives. The quickest achievement of a satisfactory smooth operation (which would probably also tend to maximize successful instructional use) occurred at institutions where (1) prior faculty/staff experience with computing existed in some numbers (10% to 20% of faculty prior to arrival of the minicomputer, and (2) ample time was provided to allow at least half his time to develop the facility.

Question 7: What are recommended steps in computer selection to assure a reasonable choice of equipment?

Response: If satisfactory computing service is a primary objective, don't be a pioneer. One institution's hints were: (a) buy from an established vendor. (b) Select a time-tested mainframe and peripherals. (Personal visits and calls on customers with the same equipment are strongly advised.) (c) Stipulate in the contract that all hardware and software must be in full operation on-site before any payment is made. Another institution suggested asking vendors for sample systems to run experimentally for a test period. Still another suggested that primary attention should be paid to the availability of time-tested software in the areas of most interest.

Proposed New NSF Support Program

The National Science Foundation is currently considering a new program which has a catchy acronym,namely, MICAT. It stands for "Minority Institutions Computing and Technology.

The emphasis of MICAT is on the development of high quality, culturally specific, computer-based instructional programs. The program takes into consideration the fact that the learning processes for minority group members are different from Anglo-Americans. Thus, the primary thrust of MICAT is on the development of specialized courseware and instructional techniques tailored to specific educational needs and cultural characteristics of minorities. The minorities of concern are Native Americans, Blacks, Hispanics, Asians, geographically isolated Americans, and others which are under represented in scientific careers.

MICAT purports to provide support for the following activities:

1) faculty training programs in computer use technology to the extent that they can adequately select equipment, modify existing courseware, and develop new culturally revelant courseware;

2) dissemination activities which include the establishment of specialized courseware libraries for use by other similar minority institutions and a visiting scientist program to further support faculty courseware development activities;

3) development of well conceived programs of instructional computing on the campuses of Minority Institutions which include both science and non-science majors;

4) provide minimum but adequate computing capability for instructional computing and courseware development efforts;

5) conduct evaluation activities which will focus on the impact of integrating computer techniques into the instructional programs of each group of Minority Institutions independently.

The training and dissemination activities will be supported through lead institutions, or equivalence, which will serve as Centers for Technological Resources (CTR's) for the other similar minority institutions. The other activities will be supported through individual grants to Minority Institutions where appropriate.

Additional information on MICAT will be made available when, and if, the NSF makes it's decision in this regard.

References

ENGEL: Final Report on the College Consulting Service of the Association for Computing Machinery, National Science Foundation Grant GY-7052.

SWOYER: "Use of Minicomputer Facilities for Higher Education Instruction", Southern Regional Edcuation Board, Atlanta, 1975, NSF Grants GJ-1072, 1111, 1116-22, 1139-40.

HANDBOOK X: EDUCATIONAL TECHNOLOGY

Ivan N. Seibert, National Center for Educational Statistics

Susan Hawkins, Association for Educational Communications and Technology

A. Lee Hall, Virginia Commonwealth University

James Mitchell, Iowa Department of Public Instruction

In an era in which special attention is being focused on alternatives to traditional modes of education, much too little is known or agreed upon concerning the role and impact of educational technology. Indeed, even the parameters of and within the area have not been clearly identified and defined. Reliable and valid statistics on the availability and use of educational technology are presently lacking or open to question. In *To Improve Learning*, (1970), the Report of the Commission on Instructional Technology to the U.S. Congress and the President, attention was called to the lack of reliable and valid data on the use of the technologies of instruction, the absence of "hard" cost data on the installation and maintenance of television systems in the nation's public schools, and the lack of up-to-date statistics on "Instructional Television Fixed Service" (ITFS) and closed circuit systems.

A major barrier to collecting, analyzing, and reporting reliable information regarding educational technologies is the lack of clearly defined concepts and standardized terminology within the field. The National Center for Education Statistics, State education agencies, and other agencies collecting statistics have been consistently handicapped by such lack of standardization. Without commonly understood and accepted concepts and terminology it is difficult to communicate adequately, to analyze data being collected, and to provide meaningful information about the field.

Scattered information in the field of educational technology--perhaps in large quantities--currently exists; but to be fully useful, it must be assembled, evaluated, and standardized. This requires extensive exploration of the sources and kinds of such data, consultation with authorities in the field, and investigation regarding particular needs and activities of various associations, organizations, and agencies actively utilizing or concerned with educational technology.

In recognition of the need for a handbook of terminology, definitions, and units of measure in this field, the Association for Educational Communications and Technology (AECT) was awarded a contract in June 1973 by the National Center for Education Statistics (NCES) for the development of such a handbook. The contractor established a national planning group, consisting of a representative from each of 19 associations or agencies, to serve in an advisory capacity and to review the materials as they were developed. The list of groups and organizations selected to review the draft informally adds more than 25 other names to the list of participating agencies.

The second draft was reviewed also in 10 regional meetings (one in each HEW regional center) in the fall of 1974 by representatives of the 19 associations and agencies from the respective States and outlying areas. The final manuscript will reflect recommendations and suggestions made by the participants in the regional and national meetings.

The handbook is based on the assumption that educational technology is *more* than "hardware" or "software." Rather, educational technology is seen as the process whereby learning resources are planned, produced, utilized, and evaluated. Thus, it is not the use of television, per se, which makes Sesame Street an example of educational technology. It is the specification of objectives, the careful analysis of learner and subject matter, the subsequent production and delivery of instruction, which, when taken together, indicate the application of technology. The final product will offer a definition of educational technology: a model of the field: a corresponding classification scheme that lists major areas of the field and their related terms; and finally, definitions for the terms in the classification scheme and for other related terms.

This handbook will be the tenth in the State Educational Records and Reports Series which includes other handbooks dealing with areas such as financial accounting, student accounting, property accounting, staff accounting, and State education agency information. The handbook series is designed so that the various handbooks are interrelated and the educational technology handbook will further extend the interrelationship of the series by providing an overall model which incorporates the subject matter of previous handbooks. Publication of Handbook X is scheduled for the summer of 1975.

ON THE PROBLEMS OF TEACHER PREPARATION IN INFORMATION SYSTEMS

T.C. Ting
School of Information and
Computer Science
Georgia Institute of Technology

Computer and information technology applications in organizations have impacted the practice of management significantly. Many information-related positions in organizations demand that formal educational programs produce high-quality information systems specialists to fulfill information-related functions. One of the deficiencies in educational programs in information systems is the lack of quality teachers. The problems of information systems in organizations and the causes for the lack of teachers in educational programs are analyzed. Preliminary suggestions are presented as a framework for the preparation of teachers in information systems programs.

INTRODUCTION

Technological advancement in information processing has affected the practice of management significantly in most organizations. Computerized information systems are being introduced into more and more organizations to automate many of the functions of information collection, storage, processing, and dissemination. These systems have demonstrated their capabilities both for reducing information processing costs and for providing better information for operations and decision-making. As the relationship of information systems to managerial effectiveness becomes more evident, the further development of information systems is promoted. The need for qualified personnel, both managerial and technical, is obvious, and the number of educated information specialists at all levels must be enlarged to meet the increasing demands for information-related positions created in organizations.

Educational opportunities in information processing systems are now available in a number of diverse programs oriented to management, to computer science, to operations research, etc., but these educational programs are not adequate to meet the quality as well as quantity needs of information personnel in industries. Improvement may be achieved by incorporating the body of knowledge in information systems into existing curricula, and by increasing research efforts concerning the design and implementation of large-scale information systems. Professional programs in information systems analysis and design should be further encouraged to transmit the growing body of knowledge to those involved with information systems.

The lack of quality teachers is a serious problem in educational programs in information systems (Teichroew, 1971). The most significant educational need at the present time is the training and preparation of teachers in the field of information systems so that these teachers can educate future practitioners, researchers, and teachers. To satisfy the ultimate educational goal in information systems, serious attention should be devoted to up grading teacher qualifications.

In this paper, the causes for the lack of quality teachers are analyzed with the hope that the result of the analysis may provide a framework for further development of teacher preparation programs in information systems. In addition, the paper discusses the issues of teacher preparation in information systems, and offers some suggestions and recommendations.

INFORMATION SYSTEMS IN ORGANIZATIONS--THE PROBLEM

Computerized information systems play a vital role in today's organizations. Operating-level administrators need information for guiding and directing organizational activities, and high-level management needs information for performing its strategic functions of controlling, planning, and development. Information systems have always existed within the traditional management functional areas. However, the rigidity of these functional divisions has been considerably impacted by the advent of computing and information technology.

Computers offer the opportunity to formalize, systematize, and automate many information-processing functions. By doing so, they encourage centralization of decision-making in that they automate routine decision-making activities and place many of the information production and dissemination functions into an integrated data base system or network. Thus, the use of computers raises not only technological questions, but also questions regarding management philosophy, regarding authority and responsibility, and regarding organizational structure. It also brings into focus the need for reviewing management policies for selecting routine decisions for automation and for validating procedures for automated decision-making and control functions.

Computing and information processing methods are often viewed as highly quantitative and technical, and since many managers do not have a clear understanding of the capabilities and limitations of information systems, they often leave the handling of information problems to technicians. They are not really very involved with those systems, nor do they really understand the effect of those systems on the organization.

Since information systems are merely tools (tools designed to serve their users), users needs must be clearly understood before a truly, effective system may be developed. Certainly, managers will need a general understanding of computers, but this understanding must be used primarily to specify the information requirements for the performance of their particular functions and responsibilities; for example, managers should be able to assess the validity of the information produced by the systems, and not be victimized by misinformation. The question then is: What level of knowledge concerning information systems must managers have in order to perform their roles in organizations of today and tomorrow?

There are in most organizations a large number of information-related positions which have no clear job descriptions. The roles of these personnel are not clearly understood by management, and supervisors sometimes simply do not know what to expect from these people even in the way of routine performance (let alone what to hope for in the way of an outstanding display of imagination and skill). The problem is due to the novelty and rapid change of the field. In many cases, individuals are occupying information-related positions for which they are inadequately qualified. Deficiencies are often due to the increasing technological requirements and managerial involvements of these positions. The issue is, To what degree should these positions be analyzed so that their major functions can be identified and classified to permit discussion of educational requirements for those now involved in the positions and for those becoming involved in the future?

On the other hand, since the technical people often do not comprehend the complexity of the organizational environment in which the information systems is to operate, they tend to underestimate the difficulty of implementing a new system. The problems in the administrative applications of computers are large-scale and ill-structured, and go much beyond merely technical considerations. However, formal academic programs tend to train their students in solving only well-structured, mathematically representable problems in the area of computing and information science. As a result, the technically oriented graduates find that organizational data processing problems seem trivial but that their environment is much more complex than that of academic-oriented problems. Therefore, one of the important questions is: How much management education should information technologists have in order to perform well in the organizations which will employ their talents?

Due to the rapid change in computing technology, the preparation of better qualified technical personnel is mandatory in order to maintain a systems stability that will persist through changes in hardware, software, and new development. On the other hand, the systems which are designed must have high flexibility in order to be responsive to changes in organizational requirements and to facilitate the making of needed system modifications in an orderly professional fashion. Another question then is, To what degree should educational programs in computer and information systems provide their students with the basic principles and foundations, so that they may cope with the problems of the fast advancement of the field and continuously progress in their technical and professional development?

EDUCATIONAL PROGRAMS IN INFORMATION SYSTEMS

How formal education programs in information systems can contribute to the effective application of computers in organizations is a particularly interesting pedagogic question. Educational programs relevant to positions in information systems are now available in several related fields.

Most business schools or departments of operations research offer some courses which introduce computers for management. However, most of there programs merely introduce the students to programing skills (McKenney and Tange, 1971). Increasing attention is being paid by some educators to develop courses so that the computing and information technology can support other management courses. Such courses are intended to expose the prospective manager to the computer.

Management education for information systems specialists is just beginning in an organized fashion (Ashenhurst, 1972). A few institutions have implemented such programs, and they are beginning to draw support from the business community.

Formal computer science degree programs are increasing in numbers at all levels of the higher educational system. These programs range from doctoral to associate degrees. Nevertheless, most of these programs are mathematically and technically oriented, and the applications of computers in organizations are thinly covered.

These programs are not adequately fulfilling the needs of the organizations. The basic problems are probably due to the following factors. The academic institutions encourage the study of well-structured problems. Most of the developmental work in information systems is done by industries and computer manufacturers. Published literature on information systems is often of inferior quality. There are no acceptable standard descriptions of information-related positions existing in terms of their function, knowledge, experience, orientation, and skills, which would allow the study of educational requirements. Above all, the most serious problem is the lack of qualified faculty members in these programs.

THE CAUSES OF THE LACK OF QUALIFIED TEACHERS IN INFORMATION SYSTEMS

The basic objectives of educational programs in information systems are to provide necessary knowledge and skills to individuals for information-related positions and to facilitate students with a sound foundation for continuing technical and professional development.

Individuals with unified knowledge and techniques in problem-solving relating to people, computational systems, and organizations are probably better equipped for the tasks of computer applications in organizations. Educational leaders have been working deligently in an attempt to develop aopropriate curricula in information systems with both technical and managerial considerations to fulfill the desired educational objectives. Since there exists no acceptable standard descriptions for information-related positions in organizations, and since the field of computing and information technology is a relatively new and fast-growing discipline, the analysis of the educational requirements for these positions is indeed a non-trivial taks. Several reports are now available for guiding the design and implementation of the educational programs in information systems and their related areas (Ashenhurst, 1971). However, the controversial curriculum issues of theoretical vs. practical approaches in computer and information science have not yet been resolved; though these issues have been widely discussed, few agreements have been reached (Kandel, 1972, and Blount and Fein, 1973). Some teachers in information systems are confused and many of them are simply taking a strong stand in the theory vs. application dichotomy. Others may speculate on the nature of the positions without real experience and knowledge of the context of the specific duties, functions, responsibilities and authority, constraints and limitations imposed on those positions.

One of the difficulties in attracting qualified teachers to educational programs is the reward system in academic institutions which is usually inclined to discourage faculty members to devote effort toward application-oriented computing and information studies. Faculty members are channelling their energy and talents toward well-structured, small-scale and theoretically oriented problems in computer and information science. It is obvious that the result of this discrimination has been to make the applied fields of computer and information technology an unpopular area of study.

The majority of research and development work in information systems are done by industries rather than universities. Industrial organizations usually do not emphasize the publication of research results, partly due to the lack of recognition of the value of publications and partly due to the intention of management to avoid offering competitive advantages to other organizations. Instead of simply communicating research findings, publications from industry often have different purposes. In some cases, such publications are commercially oriented and offer very little value as resource material for instruction and research. Faculty members frequently find themselves facing a large quantity of published literature but not enough quality references for their instructional uses or their research endeavors. Publications in the areas of applied administrative information systems are often of inferior quality and often lack detailed technical information. These publications may sometimes be used to support the teaching of introductory courses, but they do not adequately support applied courses beyond the introductory level.

Research conducted in universities on the subject of information systems is usually not too successful either. In addition to the unfavorable bias of the faculty reward system against solving applied problems in an academic environment, the students in computer and information science have been also discouraged for undertaking research in this direction. Furthermore, in addition to the lack of quality references, universities often also are not equipped qith adequate facilities for undertaking research in the areas of ill-structured, large-scale, relatively complex applied problems in information and computer science. Some computer science departments are now encouraging research in this direction, but the results are far from significant. However, this trend should be supported.

The majority of faculty members now teaching information systems courses are basically from the computer science discipline. They may have adequate concepts and knowledge in the field of computational systems, but most of them have no or little experience involving actual information systems work. They do not appreciate, nor are they interested in, the problems of computer applications in organizations. Due to the discriminatory practices of academic institutions against applications-oriented faculty, most qualified and experienced information systems specialists have no intention of leaving industry for working in instruction and research and academic institutions.

PRELIMINARY SUGGESTIONS ON THE PREPARATION OF QUALITY TEACHERS IN INFORMATION SYSTEMS

General Considerations

The following set of preliminary suggestions concerning the preparation of quality teachers in the area of information systems is based on the result of the analysis of the problems presented in the previous section. The suggestions take into account the following considerations.

1. The field of information systems mentioned here would include general topics in hardware/software components of computational systems; systems concepts and models; the organizational activities, functions, structures; and people (with emphasis on applications). Teachers of information systems may be involved in different disciplines such as computer and information science, management, operations research, etc. They are interested in a systematic body of principles and practices for large-scale information systems, but they play different roles in different programs with somewhat different objectives.

2. The preparation of quality teachers must not only consider the pre-service/in-service training programs of the teachers; it should also consider such broader subjects as curriculum in information systems and their related areas, the instructional resource requirements, student involvement in instruction and research, the general support of the teacher and of his professional, research, and intellectual development.

Suggestions

It is the author's intention to provide a set of general recommendations which may be used as a framework for developing teacher preparation and improvement programs in information systems.

1. Professional societies, industries and academic institutions interested in information systems must continue to encourage cooperative teamwork in an attempt to identify and develop educational requirements and specifications for information-related positions in organizations. Based upon the results of these studies, appropriate educational programs may be developed, and consequently the qualifications of the faculty in those programs can also be better understood. Due to the rapid changes both in computing and information technology and in management requirements, this type of study should be a continuous task.

2. Computer scientists in academic institutions must encourage application-oriented studies in order to counterbalance the theory vs. application dichotomy. We must do our best to bridge the gap between theory and practice in computer and information systems. The computer discipline may be viewed as an applied science with a theoretical foundation to support it. Basic research is important in terms of discovery of new concepts and principles in computing and information, but to continue to ignore the application of these concepts makes no sense. The solutions [illegible]plied computing problems may not be as elegant as are the answers to theoretical questions, but there are many other challenges and rewards in problems other than basic research issues.

3. Both academia and industry should promote the exchange of faculty members and information specialists in industries, in an effort to develop high-quality faculty talents. Faculty may gain first-hand experience in real organizational environments dealing with real problems instead of hypothetical issues. At the same time, information specialists in industries not only will learn new principles and new discoveries in frontier research projects in the academic institutions, but their empirical knowledge and skills will be an important contribution to faculty and students in those programs. Industries in the exchange programs will also gain substantial benefits from the talents of these exchange faculty members. In the long run, they will eventually be rewarded by the infusion into the industrial environment of high-quality graduates of formal educational institutions staffed by a high-quality faculty with an appreciation for real problems and solutions.

4. Academic programs in information systems should encourage students to participate in work-study programs. Students should be allowed to integrate the theoretical fundamentals with practical considerations for an early establishment of a balanced attitude toward theory and applications. These exchange students may bring into classroom discussion current issues related to their work-study program, and this may stimulate faculty members to be constantly aware of present problems and recent developments in the field. Experience gained from these exchange programs will not only affect the future success of the individual student himself but will also contribute to the development of faculty as well as other students, by sharing the fruitful results.

5. Industries and academic programs should promote cooperative research projects in information systems. Fundamental research and generalized systems analysis and design techniques may be best-suited for study in educational institutions. Industries should make funds and equipment to academic institutions for conducting research problems of a more general nature. Industries will obtain great contributions from the collective intellegence in universities. Faculty members will not only enjoy the satisfaction of pursuing their real research interests, but the outcomes may eventually change the reward system of the academic institutions for the encouragement of applied research effort.

6. Computer and information scientists and professional associations should promote quality publications for research and development results in the field of information systems. They should establish high publication standards and critical reviews. This will not only result in fruitful teaching references for formal educational programs but will also promote high research productivity by sharing results and ideas and thus avoiding duplications.

ACKNOWLEDGMENT

The author wishes to thank Mr. John Gehl for his helpful comments and suggestions on this draft.

REFERENCES

Ashenhurst, R.L., Ed., A Report of the ACM Curriculum Committee on Computer Education for Management, Communications of the ACM, Vol. 15, No. 5, May, 1972.

Blount, S. E. and Fein, L., The Practical Aspect of Computer Science Education--Discussion, Communications of the ACM, Vol. 16, No. 1, Jan., 1973.

Chiaraviglio, L., and Gehl, J., Some Notes on the Planning of Research in the Information Disciplines, Proceedings of the ASIS Annual Meeting, 1974, Vol. 11. Atlanta, Georgia, 1974.

Goldberg, W., On the Questions of Systems Education at Business Schools, I.G.A. Journal, Vol. 2, No. 1, Jan., 1969.

Hamblen, J. W., Using Computers in Higher Education: Past Recommendations, Status, and Needs, Communications of the ACM, Vol. 14, No. 11, Nov., 1971.

McKenney, J. T. and Tonge, F. M., The State of Computer Oriented Curricula in Business Schools 1970, Communications of the ACM, Vol. 14, No. 7, July, 1971.

Ross, J. E., Data Processing Training for Managers: Objectives and Curriculum Content, Computers and Automation, Sept., 1968.

Slamecka, V. and Zunde, P., New Educational Trends in Information Science, Proceedings of the Twenty-First Annual Meeting and Convention, New York, New York, National Microfilm Association, 1972.

Teichroew, D., Ed., Education Related to the Use of Computers in Organizations, Communications of the ACM, Vol. 14, No. 9, Sept., 1971

Ting, T. C., On the Need of Computer Education for Management, Int. J. Math. Educ. Sci. Technol., Vol. 5, No. 4, Dec., 1974.

A NEW APPROACH TO TEACHING COBOL

Joyce M. Forrest
Virginia Commonwealth University

Teaching COBOL should be a snap--at least that is the opinion of many first-time COBOL teachers. After all, many teachers either have recently graduated from a teacher-training institution where they learned to code intricate programs or have previously worked as programmers in industry where they gained their experience in manipulating the COBOL language. Therefore they know the language and feel confident in it.

The first semester that a data processing instructor teaches COBOL he invariably concentrates on the syntax of the language and the physical structure of the computer. For some reason many teachers feel that a student must have a basic knowledge of the machine before he can write an acceptable first program, although this beginning exercise is usually copied from the teacher's coding. Therefore, the initial class meetings may be devoted to discussing the input and output units, the central processing unit, and the advantages and disadvantages of COBOL when compared to other high level programming languages. Following this introduction to the computer and to computer languages, the teacher may begin a detailed examination of each of the COBOL divisions and the various required and optional sections, paragraphs, and statements within each division.

After these beginning class sessions, the class usually begins to code its first program, which may be an 80/80 listing of data cards. During this exercise the teacher will normally explain in detail each of the statements of the program as well as the optional sections and paragraphs which are available but not necessary in this program. The students take notes on the syntax of the language as they copy the coding. Although several class periods of coding may pass, the student will not be allowed to test his program yet, for he must obtain the job control cards necessary to run his job on this particular computer. As he goes through these final steps in preparing his job for compilation, the instructor may attempt to explain the function of the job cards and the manner in which the operating system interacts with the applications program. These additional lectures may require another week of class time before the student is allowed to test his program.

Needless to say, by the time the student keypunches and compiles his job he is completely overwhelmed by the mass of details presented to him during these preceding weeks. His enthusiasm for learning "data processing" has been squelshed. He has become totally confused with the numerous rules and statements associated with the COBOL language. For the duration of the year the teacher will never be able to rekindle the interest and enthusiasm which the student initially displayed. But even more importantly, the student has probably developed a dislike for COBOL--a dislike generated because he feels that a proficiency in this language is unattainable for him. Yet, this is the one business data processing language that most programmers are required to know on the job.

Some instructors may vary their approach to teaching COBOL. During the initial weeks of class many will place less emphasis on the computer technology in an attempt to present fewer machine-related facts so the student may test his programs earlier in the semester. Yet, without realizing it the teacher may still be spending too much time in these weeks of class discussing and explaining COBOL instructions rather than allowing the student to code and test his program. Until the student has actually used the equipment and processed several jobs, much of this discussion will have little value for him.

Now there is a totally new approach to teaching COBOL--an approach which allows the student to test his program on the first or at most the second day of classes. First, all the optional words, paragraphs, and sections of the program must be culled out for this introductory job. Then, by using the low-volume input and output statements, the file descriptions and record formats will be eliminated. The initial programming exercise, though relatively useless to an on-the-job programmer, provides the beginner with an opportunity to carry a computer job through all the required phases between coding the instructions to generating the output. However, the quantity of material presented is not so great as to make him feel uncomfortable.

This first programming assignment contains the simplest types of COBOL instructions (See Exhibit A). There are no superfluous statements within the exercise. In fact, there are no sections within the Environment Division and only

```
TIME=12:14:49    VERSION=01FEB74     *****  WATBCL  *****  ID=VCUC  DATE=THU, FEB. 27,

          $JOB    ABC6    WATBCL  FORREST

          001010 IDENTIFICATION DIVISION.
          001020 PROGRAM-ID.   PROG1.
          001030 ENVIRONMENT DIVISION.
          001040 DATA DIVISION.
          001050 WORKING-STORAGE SECTION.
          001060 77  CARDIN             PICTURE X(80).
  1       001070 PROCEDURE DIVISION.
  2       001080      ACCEPT CARDIN.
  3       001090      DISPLAY CARDIN.
  4       001100      ACCEPT CARDIN.  DISPLAY CARDIN.
  6       001110      ACCEPT CARDIN   DISPLAY CARDIN.
  8       001120      STOP RUN.

017347924JANET L RICHTMYER      4101 WELFORD AVE        RICHMOND VA         232340
021365488KAREN J WINSTEAD       10309 CONFEDERATE LA    FAIRFAX VA          220301
036280759JOANNE F COOMBS        1208 MAYWOOD RD         RICHMOND VA         232290
```

EXHIBIT A

one statement under the Identification Division. There are no paragraphs within the Procedure Division; and since the program utilizes no files or records, there are no Open, Close, Read, or Write verbs required. Since this exercise produces an 80/80 listing without fillers between fields, group and elementary levels are unnecessary in the Data Division and only one data-name is required.

Although the coding is unquestionably simple and straightforward, this easy exercise will lead to discussions of many more complex programming techniques. For example, because three individual Accept and Display verbs are executed in order to read and print three data cards, a student may question the manner in which multiple records will be handled. Thus may emerge the technique of utilizing loops and branches, the necessity for using procedure-names, and may ultimately lead to a discussion of modularizing a program. The Read verb with the At End option may produce a discussion of sequential file manipulation and the advantages and disadvantages of this type of organization.

When the class begins looking at the report which the program generated, undoubtedly someone will question why there are no spaces between the different fields. The teacher may then alert the students to the manner in which COBOL handles files, records, and fields. There follows a thorough examination of group and elementary levels found in the Data Division and the various types of picture formats which may be used to define the data in the fields. After completing each element in the Data Division, the teacher may then proceed to the Select statements in the Environment Division and the high-volume input and output statements in the Procedure Division. In this manner the student can distinguish the relationship among the different COBOL divisions.

Thus, with this initial programming exercise the student has the experience of coding, compiling, and debugging a problem. Each of these steps presents something new to the student. The quality of new material is not so overwhelming as to discourage his initial attempts of learning a computer language. Instead of being disheartened, the student will accept the concepts with more open-mindedness, and mastering the language becomes less difficult. From this elementary beginning the student develops a second program almost identical to the first one, except that this one uses the high-volume input and output verbs and a multiple number of data cards. The same input records are read by this second program as by the first program; there are just more of them this time. This second exercise builds upon the first, adding paragraphs, records, and files to the basic structure. (See Exhibit B)

Whereas the first exercise was simply copied from the teacher's coding, this second exercise should allow the student to program the problem himself. Perhaps the instructor should allow the student at least one laboratory day for working on this program during class time; in this manner the teacher will be available for individual consultation when programming problems arise. This second exercise should be accompanied by a detailed flowchart from each student. Although the flowchart requires few decisions to be made, this pictorial diagram will give the student the experience of drawing a problem solution and then using this solution to solve the problem in the COBOL language. The teacher should begin emphasizing the flowchart and documentation requirements early in the course so that the student becomes accustomed to them and assumes that they are naturally a part of programming itself.

Regardless of how hesitant the student may be to commit himself to flowcharting and programming this second job, the teacher should encourage him to be on his own as much as possible. Although the student may have to spend additional time debugging this job, he will gain more confidence in himself and be more willing to tackle the next assignment if he feels that he has mastered this one individually. Should the student not be encouraged to assume this responsibility at this time, he will become more and more dependent on his instructor and will never feel successful in using COBOL.

This second program will probably be completed during the second week of classes, and by the beginning of the third week the student will have already written and tested two programs. Thus he will have a feeling of accomplishment at the very start of the class. His interest and enthusiasm will not be stifled as they had been previously. Thus, the teacher does not have to worry about rekindling that enthusiasm as he attempts to expand into more complex programs and more intricate programming. Because the student can see his progress during the past several weeks, he has a feeling of self-confidence which encourages him to begin the next assignment.

Each COBOL exercise which follows these first two should be slightly more difficult than the previous one. Perhaps by the third exercise the student is ready to begin writing headings on the report and comparing codes in the data. Gradually the instructor should introduce the more advanced concepts of multiple input/output files, the sort verb, multiple counters and master file updates. Each exercise should be reasonably short and concise and should introduce one or more new techniques. The teacher should not restrict jobs to card input and printer output if additional input and output devices are available. Also, the teacher should promote business-type applications throughout the course, emphasizing such everyday business requirements as proofing the input data for errors before processing them and accounting for each record read and handled during a job.

Some students will always have difficulty in mastering a programming language, and these students should perhaps be encouraged into some other major. However, for the majority of students this simple approach to learning the COBOL language provides an opportunity to find success rather than discouragement. Perhaps the student is more willing to continue from one program to the next because the quantity of new material he must understand is not so great as to be overpowering. Each time the student is successful in his programming attempt, he is encouraged to begin the next assignment. The instructor who uses this approach to teaching COBOL provides the student with the opportunity to be successful from the beginning of the course by introducing the new material to the student in manageable quantities.

The teacher who uses this approach is better equipped to locate trouble spots when they occur. The teacher may discover that the students have more difficulty in reasoning through their problems than they have in learning the syntax of the language. Thus, the programming teacher may wish to abandon COBOL instruction for several class meetings in order to concentrate on flowcharting and problem-solving techniques in more detail. Thus the instructor is better able to distinguish between coding difficulties and problem-solving difficulties.

Until a student registers for a programming class he may never have seen or used a computer; yet, without knowing anything about the subject, he is somehow motivated to learn "data processing." Thus, the teacher's main objective should be to have each student compiling his programs on the computer as soon as possible. By allowing the student to code and test during the initial week of classes the teacher stimulates his interest and enthusiasm in learning the subject. While all his other classes are demanding him to read more and more textbook chapters, the student is able to do something in his COBOL class. Immediately he has a good feeling for data processing and is more willing to devote the time and energy necessary for the class. The teacher, too, will reap the rewards of a first-day COBOL program, for since he never allows the students to lose their initial enthusiasm he does not have to struggle to bolster class morale.

The student feels successful in programming rather than overwhelmed by an enormous mass of detail and statements. He tries to master one thing--COBOL. Later he will attempt to understand operating systems and computer mechanisms. He becomes comfortable with the language immediately upon introduction to it. He is more willing to persevere through the numerous compiles and tests because he is confident he can succeed; and, hopefully, this confidence will lead him unhesitatingly into more complex programming techniques with as much success as he has experienced in his first attempt.

```
TIME=15:57:37    VERSION=01FEB74      *****  WATBOL  *****  ID=VCUC  DATE=THU, FEB. 2

        $JOB   ABC6    WATBOL  FORREST

        001010 IDENTIFICATION DIVISION.
        001020 PROGRAM-ID.  PROG2.
        001030 AUTHOR.  STUDENT NAME.
        001040 DATA-WRITTEN.  SEPT. 4, 1974.
        001050 REMARKS.  PROGRAM PRODUCES A STUDENT NAME AND ADDRESS FILE.
        001060 ENVIRONMENT DIVISION.
        001070 CONFIGURATION SECTION.
        001080 SOURCE-COMPUTER.  IBM-370.
        001090 OBJECT-COMPUTER.  IBM-370.
        001100 INPUT-OUTPUT SECTION.
        001110 FILE-CONTROL.
        001120      SELECT FILEIN ASSIGN TO UT-S-SYSIN.
        001130      SELECT FILEOUT ASSIGN TO UT-S-SYSOUT.
        001140 DATA DIVISION.
        001150 FILE SECTION.
        001160 FD  FILEIN  LABEL RECORDS ARE OMITTED RECORDING MODE IS F
        001170      RECORD CONTAINS 80 CHARACTERS DATA RECORD IS STUDENT-ID.
        001180 01  STUDENT-ID.
        002010      03  SOCIAL-SECURITY          PIC X(9).
        002020      03  NAME                     PIC X(22).
        002030      03  STREET                   PIC X(24).
        002040      03  CITY-STATE               PIC X(19).
        002050      03  ZIP                      PIC X(5).
        002060      03  KODE                     PIC X.
        002070 FD  FILEOUT LABEL RECORDS ARE OMITTED RECORDING MODE IS F
        002080      RECORD CONTAINS 133 CHARACTERS DATA RECORD IS PRT-REC.
        002090 01  PRT-REC.
        002100      03  FILLER                   PIC X(6).
        002110      03  SOCIAL-SECURITY          PIC X(9).
        002120      03  FILLER                   PIC X(6).
        002130      03  NAME                     PIC X(2[illegible].
        002140      03  FILLER                   PIC X(6).
        002150      03  STREET                   PIC X(24).
        002160      03  FILLER                   PIC X(10).
        002170      03  CITY-STATE               PIC X(19).
        002180      03  FILLER                   PIC X(6).
        002190      03  ZIP                      PIC X(5).
        002200      03  FILLER                   PIC X(9).
        002210      03  KODE                     PIC X.
        002220      03  FILLER                   PIC X(8).
  1     003010 PROCEDURE DIVISION.
        003020 BEGIN.
  2     003030      OPEN INPUT FILEIN OUTPUT FILEOUT.
        003040 MAIN-ROUTINE.
  3     003050      READ FILEIN AT END GO TO FINISH.
  5     003060      MOVE SPACES TO PRT-REC.
  6     003070      MOVE CORRESPONDING STUDENT-ID TO PRT-REC.
  12    003080      WRITE PRT-REC AFTER ADVANCING 2 LINES.
  13    003090      GO TO MAIN-ROUTINE.
        003100 FINISH.
  14    003110      CLOSE FILEIN, FILEOUT.
  15    003120      STOP RUN.
```

EXHIBIT B

```
036280759      JOANNE H [illegible]            1208 MAYWOOD RD                RICHMOND VA             23229      0
069364716      ARLENE H UNGER                  4112 I TOWNHOUSE RD            RICHMOND VA             23228      0
090122377      ANN R PRINCE                    193[illegible] WESTOVER HILS BLVD   RICHMOND VA        23225      5
094402033      KATHLEEN M ALLEN                189 FREDERICK ST               PEEKSKILL N Y           10566      0
096362506      DIANE L CHAMBERS                SCOTCH GROVE APTS              PULASKI N Y             13142      0
099128854      FRANK J DROPELLO                3140 LENEWOOD DR               RICHMOND VA             23234      0
110302260      DENNIS J PASQUANTINO            3553 MEADOWDALE BLVD           RICHMOND VA             23234      0
111302894      BRIAN KANE                      2807 SUNORA CT                 RICHMOND VA             23234      0
146201776      ANN S RICHARDSON                BOX 452 NWL                    DAHLGREN VA             22448      0
149334336      CATHY A NEMCHIK                 167 FETTER AVE                 TRENTON N J             08610      0
161401654      JANICE L BRINDLE                RD #3 SILOAM ROAD              CHAMBERSBURG PA         17201      0
171140276      MARY G LANCASTER                3011 ELMBROOK RD               RICHMOND VA             23228      0
193387339      MARGARET L WINDISCH             105 WINSTON AVE                COLONIAL HEIGHTS VA     23834      1
LE TIME        0.45 SECONDS TASK;      5.13 SECONDS REAL
TION TIME      0.04 SECONDS TASK;      1.19 SECONDS REAL
 READ DURING COMPILE           53 DURING EXECUTION       16
 PRINTED DURING COMPILE        56
 PRINTED DURING COMPILE         1 DURING EXECUTION        1
AVAILABLE AT ENTRY TO EXECUTION 77016 BYTES
```

EXHIBIT B

TRAINING PROGRAMS FOR SYSTEMS ANALYSTS

Harrison S. Carter
Virginia Commonwealth University

Introduction

This paper describes the results of a project undertaken by the Division of Automated Data Processing (ADP) of the State of Virginia.[1] The primary purpose of the project was to develop a training program for practicing systems analysts within the agency. Impetus for this project grew out of a recognition by the agency that incumbent systems analysts, although well schooled in the hardware and software technicalities, required additional skills if they were to successfully perform the many interdisciplinary tasks required of the systems analyst.

This need for education and training of systems analysts is certainly not peculiar to ADP. The Curriculum Committee on Computer Education for Management (C^3EM) of the Association for Computing Machinery (ACM) has conducted a study on curriculum development for MIS in higher education. The primary results of this study to date has been detailed recommendations for graduate and undergraduate programs in information systems. These recommendations are the single most definitive work to date regarding information systems curriculum in higher education. In one of the committee's reports, they cite several problems related to the use of computers in organizations today. One of the major problem areas they focus their attention on is that of inadequate preparation. The committee states that: "personnel are occupying positions for which they are inadequately qualified, through deficiencies in computer systems and management skills, training and experience (2)." Unfortunately the prognosis for the immediate future is not good. The IBM Corporate Staff for Manpower Planning and Personnel forecasts a deficit of 62,000 systems analysts for 1975 (4). Many other studies have focused on the shortage of trained personnel to man the increasing number of systems analysts positions. While the forecasts of shortages vary, the studies do support the position that the shortage is acute (2). An inevitable consequence of this shortage will be an increasing number of inadequately prepared systems analysts. Therefore research and development in the area of training and education for systems analysts should be a high priority item for educators and practitioners in data processing. It is to this end that the research described in this article is directed.

Approach

The process of curriculum development involved several major inputs. Since the training program was designed for the practicing analyst as opposed to the trainee, the varied backgrounds and experience levels among the incumbents complicated the curriculum selection. There were ten analysts selected to participate in the training program. The background and skills of these individuals were carefully examined and constituted a major input into the process of curriculum development.

The results of this analysis indicated that the incumbents were well schooled in the hardware and software areas but were, to some extent, deficient in one or more of the following areas:

1. Information Systems Theory;
2. Management and Organization Theory;
3. Operations Research;
4. Systems Design Techniques.

These four areas are identified by C^3EM of ACM as major topic areas in formal education programs designed to produce systems analysts (2).

The identification of these areas provided a framework for the development of the curriculum. The next step in the process of developing the training program was to further define the topics to be covered by the program and to specify the detail with which each topic would be covered. This step was accomplished by George Kotlan and the writer, faculty members at Virginia Commonwealth University (VCU). Our approach was to review a number of textbooks in systems analysis and design and to study the curriculum recommendations of C^{3}EM of ACM (1). The actual contents of the training program will be covered in the next section. Following this further definition, the ADP project team investigated alternative systems analysis and design training programs currently on the market that treated those areas identified as necessary to a successful training program. This investigation resulted in the purchase of a video-assisted instruction course which became an integral part of the training program.

---

[1] Members of the project were: Dr. Dave Thomas, Mr. Sam Vaeth and Mr. Tom Hallman of ADP; and Mr. George Kotlan and the writer of Virginia Commonwealth University.

After reviewing various textbooks and the ACM curriculum recommendations regarding systems analysis and design, it became apparent that there were many legitimate topics that should be included in the program. To cover each topic in depth would have required an inordinate amount of time, therefore, it was decided to treat a fairly wide range of topics but in less detail than was desired. This approach was taken in the hopes that the incumbents would see the value and applicability of the topics covered and investigate them further on an individual basis.

When conducting training programs such as this, there are a number of outcome objectives that are possible. Some of these are as follows:

1. to impart knowledge;
2. to develop skills;
3. to change attitudes;
4. to change individual behavior;
5. to change individual behavior within the organization.

Given the number of topics that were deemed necessary and the time requirements, the outcome objective of this training program was to impart knowledge.

In summary, there were several major inputs into the process of curriculum development. These inputs included the following:

1. knowledge and background of the incumbents;
2. subject areas recognized as integral to the knowledge base desired for systems analysts;
3. available training packages;
4. number of incumbents;
5. time constraints.

Training Program

The training program consisted of a multi-media approach. The video-assisted instruction course covered only a portion of the topics i [illegible] program and was augmented with se [illegible] and individual readings. The seminars were given during the work day according to the course schedule. The video-assisted instruction and the readings were on an individual basis. Checkpoints were set up so that blocks of material would be completed by certain dates. The analysts were given an adequate amount of time during the work day to complete these individual assignments. The topics, the amount of time devoted to each, and the media are presented in Table I.

Table I

Training Curriculum

Subject	Length of Time	Media	Instructor
Introduction	2 hours	Seminar	Project Staff
System Development Process	4 hours	Seminar	Carter
User Relations	3 hours	Seminar	Kotlan
Systems Approach/ Problem Definition	3.5 hours	VAI[2]	Unit 1
Data Gathering	3.5 hours	VAI	Unit 2
Information Analysis	3 hours	VAI	Unit 3
Checkpoint Review I	2 hours	Seminar	Project Staff
Work Measurement	3 hours	Seminar	Carter
Sampling and Estimating	3 hours	Seminar	Carter
System Flow Charting	Variable	Readings	Hartman (3)
Process Flow Charting	Variable	Readings	Lazzaro (5) Systemation Vol 16 (6)
Layout Charting	Variable	Readings	Lazzaro Systemation Vol 16
Organization Structures	Variable	Readings	Systemation Vol 1
Forms Analysis	Variable	Readings	Systemation Vol 5
Documentation	5 hours	VAI	Unit 7
Checkpoint Review II	2 hours	Seminar	Project Staff
Historical Data Analysis and Forecasting	3 hours	Seminar	Carter
Cost Benefit Analysis	4 hours	Seminar	Carter
Creativity	1 hour	Seminar	[illegible]
Work Simplification	Variable	Readings	Systemation Vol 8
Procedures Design	Variable	Readings	Lazzaro Lazzaro Mathies (7)
Checkpoint Review III	3 hours	Seminar	Project Staff
Forms Analysis	2 hours	Seminar	Blanks
Report Writing	2 hours	Seminar	Hall
Presentation Techniques	2.5 hours	VAI	Unit 4
Meeting Techniques	2 hours	Seminar	Hall
Final Review	2 hours	Seminar	Project Staff
Critique	3 hours	Seminar	Project Staff

---

[2] Video-Assisted Instruction

Conclusions

The conclusions drawn in this section are based upon inputs obtained from the analysts during the training program. There were three checkpoint reviews, a final review and a final critique session during which suggestions and criticisms were solicited. The project staff enjoyed a very open relationship with the analysts. Therefore, in my opinion, we received quality, unbiased feedback from the group.

The majority of the group felt that the training program was too long and did not take into account the individual needs and interests of each analyst. Some of the analysts felt that they were not given sufficient time to complete the individual assignments. In general, the group felt that the video-assisted instruction units were too elementary and with the exception of a few of the seminars, they felt that none of the topics were covered in enough detail. From the point of view of a training coordinator, the writer feels the program had two major shortcomings. First, the program did not take into consideration the individual backgrounds and interests of the incumbents. Secondly, not enough time was devoted to any single topic so that a person deficient in that area could adequately correct that deficiency.

Accepting the assertion that there are inadequately prepared individuals occupying systems analyst positions and keeping in mind the distinction between training a trainee for a systems analyst position and training a practicing systems analyst, the writer would offer the following approach. Select a well qualified senior analyst within your organization or contract with such an individual outside your organization. Have this person evaluate on an individual basis each one of your analysts. The results of this evaluation should be an identification of the strengths, weaknesses, and current interests of each analyst. Using this analysis in conjunction with your training coordinator and the analyst, develop an individual training program that accomplishes the organization's training objectives as well as those of the analyst. The program could well make use of higher education courses, manufacturers' training courses, seminars, video-assisted instruction, on-the-job training, etc. This type of "training by objectives" approach seems intuitively more acceptable by the analyst and could provide a means of correcting the problem of inadequately prepared individuals function as systems analysts.

REFERENCES

1. Communications of the ACM, "Curriculum Recommendations for Undergraduate Programs in Information Systems," December 1973, Volume 16, No. 12.
2. Communications of the ACM, "Education Related to the Use of the Computers in Organizations," Vol. 14, No. 9, September, 1971.
3. Hartman, W., Matthes, H., and Proeme, A., Management Information Systems Handbook, McGraw-Hill Book Company, New York, 1968.
4. Hunter, G. Truman, "Manpower Resources for Management Information Systems," Procedures Founder's Conference, SMIS, September 8-9, 1969.
5. Lazzaro, Victor, Systems and Procedures, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1968/
6. Mathies, L. H., Systemation Series, (17 Volumes).
7. Mathies, L. H., Playscript Procedures.

EPIC USERS PANEL

Panelists:
Mr. S. M. Raucher, Chairman, IBM Corporation
Ms. J. E. Ellis - City of Virginia Beach
Mr. D. L. McReynolds - Oklahoma City Public Schools
Ms. A. Wood - Virginia Beach City Public Schools

The EPIC users panel is designed to promote an interchange of users experience with the four EPIC program products.

EPIC:SOCRATES	- Student Scheduling and Master Schedule Generation
EPIC:FAST	- Test Scoring and Analysis
EPIC:Student	- Mark Reporting and Attendance Accounting
EPIC:Budget/ Finance	- Accounting and Budget Control

Panelists include users who have successfully installed and utilized these systems in their schools. In addition, the most current maintenance releases will be discussed, including new options in these systems.

Ample opportunity for questions and discussion will be provided.

AUTHOR/CONTRIBUTOR INDEX

Altschuler, Henry . . . 40
Ameen, David A. . . . 115
Anderson, G. Ernest . . . —
Anderson, Gary J. . . . —
Ballard, Richard L. . . . 95
Bassler, Richard A. . . . 7
Bewley, William L. . . . 108
Blanks, Edwin E. . . . 17
Bregar, William S. . . . 89
Breitler, Alan L. . . . 118
Brumbaugh, Kenneth E. . . . 3
Bukoski, William J. . . . 1
Carter, Harrison S. . . . 143
Charp, Sylvia . . . 40
Collins, Eugene A. . . . 16
D'Orazio, Richard . . . 40
Elliott, Portia C. . . . 69
Ellis, J. E. . . . 146
Engel, Gerald L. . . . 121
Forrest, Joyce M. . . . 138
Frederick, Franz . . . 11
Fry, Charles H. . . . 11
Hall, A. Lee . . . 132
Hall, John H. . . . 83
Hamblen, John W. . . . 121
Hawkins, Susan . . . 132
Hayden, Dunstan . . . 106
Holznagel, Donald C. . . . 16
Johnson, Cecil . . . 11
Joslin, Edward O. . . . 37
Korotkin, Arthur J. . . . 1
Larson, Dean C. . . . 16
Lias, Edward J. . . . 56
Maloney, R. Kay . . . 45
Martuza, Victor R. . . . 74
Mayer, Herbert C. . . . 26
McConnell, Thomas J. Jr. . . . —
McNamara, James . . . 16
McReynolds, D. L. . . . 146
Mirsky, Michael E. . . . 48
Mitchell, James . . . 132
Morrison, Fred . . . 106
Morrissey, Charles A. . . . 30
Nelson, Robert D. . . . 64
Nold, John . . . —
Oliver, Lawrence . . . 121
Pasquino, Anne . . . —
Peelle, Howard A. . . . 69, —
Rappaport, Wanda . . . 106
Raucher, S. M. . . . 146
Rector, Robert . . . —
Richards, Thomas C. . . . 53
Seibert, Ivan N. . . . 132
Spuck, Dennis W. . . . 16
Stiglmeier, John J. . . . 48
Swearingen, John K. . . . —
Swoyer, Vincent H. . . . 121
Thompson, Norman E. . . . 16
Thornton, Jack D. . . . 78
Ting, T. C. . . . 133
Van Dusseldorp, Ralph . . . 16
Van't Roer, Jill A. . . . 41
Weinberg, David Allen . . . 107
Williams, Charles M. . . . 99
Williams, J. N. . . . 78
Wood, A. . . . 146
Worner, Roger B. . . . 21
Yens, David P. . . . 11, 74